Fast Automated Estimation of Variance in Discrete Quantitative Stochastic Simulation

November 2010

Nelson Shaw
njd50@uclive.ac.nz
Department of Computer Science and Software Engineering
University of Canterbury, Christchurch, New Zealand

Supervisor: Krys Pawlikowski
Krys.pawlikowski@canterbury.ac.nz

Supervisor: Don McNickle
don.mcnickle@canterbury.ac.nz

Abstract

Quantitative stochastic simulation is an important tool in assessing the performance of complex dynamic systems such as modern communication networks. Because of the proliferation of computers and devices that use and rely on networks such as the internet, assessing the performance of these networks is important to ensure future reliability and service. The current methodology for the analysis of output data from stochastic simulation is focused mainly on the estimation of means. Research on variance estimation focuses mainly on the estimation of the variance of the mean, as this is used to construct confidence intervals for the estimated mean values. To date, there has been little research on the estimation of variance of auto correlated data, such as those collected during steady-state stochastic simulation. This research investigates different methodologies for estimation of variance of terminating and steady-state simulation. Results from the research are implemented in the simulation tool Akaroa2.

Acknowledgements

The author would like to thank his supervisors Krzysztof Pawlikowski and Don McNickle for their support and guidance throughout the project. A special thank you also goes to Greg Ewing for his help relating to understanding and editing the Akaroa2 software, not to mention replying to so many emails!

Table of Contents

1. Introduction

Quantitative stochastic simulation is an important tool in assessing the performance of complex dynamic systems such as modern communication networks. Because of the proliferation of computers and devices that use and rely on networks such as the internet, assessing the performance of these networks is important to ensure future reliability and service. Although of performance analysis methods are available, due to the complexity and size of experiments undertaken, simulation is often seen as the preferred choice.

The current methodology for the analysis of output data from stochastic simulation is focussed mainly on the estimation of means. Research on variance estimation focuses mainly on the estimation of the variance of the mean, as this is used to construct confidence intervals for the estimated mean values. To date, there has been little research on the estimation of variance of auto correlated data, such as those collected during steady-state stochastic simulation.

Variance of delays is an important performance measure in modern networks where content is delivered in real time, and any large changes in delay of packet delivery have a significant degenerative effect on quality of service. In such cases one needs to analyse variance and its associated error, and the latter is equivalent with analysis of variance of estimated variance. This project focuses on the testing and development of a reliable, accurate algorithm for assessment of variance. The best algorithm for automated sequential analysis of variance is identified and implemented in Akaroa-2 [1]. With a better algorithm for analyses of variance, simulation can be used to test and analyse future networks and improve the performance of real time content delivery.

Although previous work in this area has been undertaken by Schmidt in [2], not all proposed estimators were fully tested. This work will be an extension of his research, focussing on additional tests to find the best algorithm.

2. Background

This part of the paper provides an introduction to the theoretical background of simulation. It explains the use of simulation in terms of modelling real world systems, and also provides an introduction into the estimation of statistics and stochastic processes.

2.1. Discrete Event Simulation

Discrete event simulation refers to the simulation of a sequence of events, where each event occurs at a specific instance in time.

Discrete event simulation is often used as a way of simulating a real world system. In order to model a real system, the first step is to create a model of the system using the methodologies and functionalities offered in the simulation environment. Using the simulation model, the simulator then runs the model using pseudo random numbers as input. The simulator then outputs observations (results of measurements of simulated processes), which are interpreted by the user to make statements about the performance of the system.

We define two different types of simulation investigated in this paper:

- *Terminating Simulation*
- *Steady-State Simulation*

Terminating simulation is defined as simulation with a specific stopping criterion. The nature of the stopping criterion is either a time or specific event. For example, a simulation model of a factory from 8am to 5pm could be analysed using terminating simulation. Because of the nature of terminating simulation, the initial state of the model can have a large impact on the results. For this reason, choosing the initial state is important.

Steady-state simulation is defined as simulation of the long-run behaviour of a system. As the true steady state of a system is defined as the behaviour as time tends to infinity, the simulation methodology used here is just an approximation. Because of the initial state of a given model, there is an initial transient or "warm-up" period between the initialization of the simulation and the realization of steady-state. It is often claimed that observations from the initial transient period bias the results of the steady state behaviour, and as such are usually discarded.

In this paper we discuss the estimation of variance in both terminating and steady-state simulation. Each type of simulation provides its own unique requirements and challenges in regard to the estimation of variance.

2.2. Estimation

As with any scientific measurement, for it to be regarded with credibility, we have to calculate according to approved scientific measure and corresponding error. Applying this principle to the results from simulation output, the results should be constructed via rigorous statistic methodology and have an associated error. For this reason, we often report results in terms of a *point estimate* and *interval estimate*. These two parameters define the region in which the true result lies and is associated with a given probability.

The estimator of parameter θ is called $\hat{\theta}$. We define the point estimate of the parameter as $\hat{\theta}$ with upper and lower bounds of the confidence interval $\hat{\theta}_h$ and $\hat{\theta}_l$. From these definitions, we can describe the confidence interval as

$$Pr[\hat{\theta}_l \leq \theta \leq \hat{\theta}_h] \geq 1 - \alpha.$$

Where $1 - \alpha$ is the *confidence level*. Often in simulation, the confidence interval can be symmetric and the *half-width* of the interval can be defined by a single parameter Δ. If this is the case, we can describe the confidence interval as

$$\Pr[\hat{\theta} - \Delta \leq \theta \leq \hat{\theta} + \Delta] \geq 1 - \alpha.$$

From these definitions of a point estimate and confidence level, we can derive measures of statistical error. There are two measures of error (or precision) we derive, *relative error* and *absolute error*. The relative error is defined as

$$\varepsilon_r = \frac{\Delta}{\theta}.$$

The absolute error is defined as the size of the half-width, that is, $\varepsilon_a = \Delta$.

Several other measurements of estimators are important in terms of determining the accuracy of the estimator. We define *bias* as the difference between the estimated value of a parameter and its true value. That is,

$$Bias[\hat{\theta}] = E[\hat{\theta} - \theta].$$

We define the *variance* of an estimator as its expected squared deviation from the mean. That is,

$$Var[\hat{\theta}] = E[(\hat{\theta} - E[\hat{\theta}])^2].$$

2.2.1. Mean Value Estimation

In regard to *mean value estimation* we are estimating the average value a parameter assumes during a simulation. If we have a mean value v, the known well-behaving estimator is the sample average. This is simply calculated as

$$\bar{X}(n) = \frac{1}{n}\sum_{j=1}^{n} X_j. \tag{2.1}$$

From [3], the variance of $\bar{X}(n)$ can be calculated as

$$Var[\bar{X}(n)] = \frac{\sigma^2}{n}\left(1 + 2\sum_{j=1}^{n-1}(1 - j/n)\rho_j\right), \tag{2.2}$$

where $\rho_j$is the auto-correlation coefficient of observation j. Using the variance, we can then calculate the confidence interval for the mean estimator $\bar{X}(n)$. We define the half-width of the confidence interval as

$$\Delta_{\bar{X}(n)} = t_{d,1-\alpha/2}\sqrt{Var[\bar{X}(n)]},$$

where $t_{d,1-\alpha/2}$ is the $1 - \alpha/2$ quantile of the S*tudent's T-distribution* with *d* degrees of freedom.

For uncorrelated observations, $\rho_j = 0$, which gives

$$Var[\bar{X}(n)] = \frac{\sigma^2}{n}.$$

However, for correlated observations, $\rho_j > 0$, we cannot calculate the variance as $\frac{\sigma^2}{n}$, because it leads to incorrect confidence intervals. Unfortunately, many queuing systems produce correlated results, which mean that special methods of estimation of the variance of the mean for correlated observations are needed. A survey of problems and related solutions to this area can be found in [4].

2.2.2. Variance Estimation

In terms of variance estimation, we are interested in estimating the variance of a parameter in a given simulation along with an associated confidence interval. The equations presented here are given in [2], and form a mathematical introduction to the estimators presented later in the paper.

If we have a given set of *independent* and *identically distributed* random variables, the accepted and well-behaving point estimate is given as

$$S^2(n) = \frac{1}{n-1}\sum_{j=1}^{n}\left(X_j - \bar{X}(n)\right)^2. \tag{2.3}$$

By [5] we can calculate the variance of $S^2(n)$ as

$$Var[S^2(n)] = \frac{1}{n}\left(\mu_4 - \frac{n-3}{n-1}\sigma^4\right), \tag{2.4}$$

where σ^2 is the steady-state variance and $\mu_4$the fourth central moment of the steady-state distribution.

In the case of correlated observations, $S^2(n)$ is biased and no longer a good estimator. This is often the case in queuing systems, and alternative estimators are described and tested later in the paper.

In terms of the distribution of $S^2(n)$, we know that if $X_i$are independent and follow a normal distribution, then $S^2(n)$ follows an χ^2 (Chi-square) distribution with $n-1$ degrees of freedom [6]. However, the confidence intervals derived using the χ^2 distribution can be quite poor when the X_i distribution differs significantly from the normal distribution. For mean values however, the construction of confidence intervals for X_i using the normal distribution is highly reliable even when the X_i distribution is considerably different from normal.

2.2.3. Evaluation of estimators

As has been discussed earlier, some valuable properties of a given estimator are its bias and variance. However, these properties to not evaluate the accuracy of the resulting point estimates and confidence intervals produced from a given estimator. For this purpose, analysing the *coverage* of the estimator is important.

2.2.3.1. Coverage Analysis

Coverage, or *coverage of confidence intervals*, c, is defined as the relative frequency that a confidence interval $(\bar{X}(n) - \Delta, \bar{X}(n) + \Delta)$ contains the true value μ_x [7]. When using coverage analysis in this research, we follow the principles and guidelines of coverage analysis presented in the paper [7]. A description of the process of coverage analysis follows.

In an ideal situation, a confidence interval result from a simulation with associated probability $1-\alpha$ would contain the true parameter in $(1-\alpha)100\%$ of final confidence intervals. However, in reality, many estimators do not produce results that obey this expectation. This is because many estimators make assumptions relating to statistical processes, which may not be true in practice.

In analytically tractable systems, where the true value, μ_x, of a parameter is known, we can assess the coverage of an estimator of this parameter by constructing a confidence interval of the coverage. This is produced by calculating the point estimate (coverage) c, and the corresponding confidence interval. The confidence interval is then given as

$$\left(c - z_{1-\alpha/2}\sqrt{\frac{c(1-c)}{n_c}}, c + z_{1-\alpha/2}\sqrt{\frac{c(1-c)}{n_c}}\right),$$

where $z_{1-\alpha/2}$ is the $1-\alpha/2$ quantile of the standard normal distribution and n_c is the number of replicated coverage experiments. This confidence interval is derived from the properties of the binomial distribution governing the number of confidence intervals containing the true value of the parameter, μ_x. The mean value of the number of confidence intervals containing μ_x is $n_c\mu_c$. Then as $(c-\mu_c)\sqrt{c(1-c)}$ tends to the standard normal distribution as $n_c \to \infty$, the confidence interval for c can be derived. Following these derivations, an estimator of a parameter μ_x is considered valid if the upper bound of the coverage confidence interval equals at least $1-\alpha$.

In terms of the creation of the point estimate and confidence interval of coverage, c, it should be run sequentially as "… no procedure in which the run length is fixed before the simulation begins can be relied upon to produce a confidence interval that covers the true steady-state mean with the desired probability level" [8] – [9]. One issue with sequential coverage analysis is that some experiments may stop after a randomly short amount of time. This occurs due to the random nature of sequential experiments, as the stopping criterion can be temporarily satisfied. Although this is a natural occurrence in many simulation experiments, if introduced here, it has the effect of adding "noise" to the estimate. For this reason, such small simulation runs are discarded to ensure the accuracy of coverage interval estimates.

In [7], it states that the following rules should be obeyed we creating coverage analysis intervals.

Rule 1 – Coverage should be analysed sequentially, i.e. analysis of coverage should be stopped when the relative precision (the relative half-width of the confidence interval) of the estimated coverage satisfies a specified level.

Rule 2 – An estimate of coverage has to be calculated from a representative sample of data, so the coverage analysis can start only after a minimum number of "bad" confidence intervals have been recorded.

Rule 3 – Results from simulation runs that are clearly too short should not be taken into account.

It is recommended that for Rule 2, the minimum number of "bad" confidence intervals observed should be 200. In [7] it was proposed to discard results from simulations which lasted less or more than one standard deviation of the mean length, but in this research we discard only those whose lengths are shorter. This analysis should take place after the minimum number of "bad" confidence intervals has been recorded. The three rules and following recommendations are followed when creating coverage analysis intervals for estimators later in the paper.

2.3. Akaroa2

Akaroa2 is an automated simulation controller developed at the University of Canterbury. Currently, Akaroa2 is used for estimation of mean-values only, but from this research its capability will be expanded to estimate variance as well.

A brief description of how Akaroa2 is used is provided for background knowledge. A user of Akaroa2 creates a model of the system they are studying by writing a C++ program describing that system. Akaroa2 provides specific interface functions in order for the model to submit observations to the controller. Akaroa2 is in charge of running the simulation, and stopping once a defined stopping criterion is met. Parameters (for example, the stopping criterion and minimum number of runs) can be set by the user before the simulation is run. Akaroa2 is capable of several different modes of analysis, and as such can handle both terminating and steady-state simulation. In order to achieve decreased simulation runtime, Akaroa2 uses the *Multiple Replications in Parallel* (MRIP) principle to run the simulations.

2.3.1. Multiple Replications in Parallel (MRIP)

Multiple Replications in Parallel is a simulation controller architecture based on statistically independent replications of a simulation on different simulation engines. A version of MRIP for mean values is described by [10]. A global analyser is used to control the simulation, with various statistically independent engines sending back results. The global analyser then combines these results and is responsible for detection of the stopping criteria. It can be shown that the expected speedup of simulation using MRIP follows a truncated version of Amdahl's law in [11]. The speedup is expressed as

$$S = \frac{1}{\bar{f} + (1 - \bar{f})/P}, for\ P \le \frac{(1 - \bar{f})\bar{N}_{min}}{\bar{D}},$$

and

$$S = \frac{\bar{N}_{min}}{\bar{f}\bar{N}_{min} + \bar{D}}, for\ P > \frac{(1 - \bar{f})\bar{N}_{min}}{\bar{D}},$$

where S is the expected speedup, $\bar{f}$ the fraction of serial (non-parallelisable) part of a program, $\bar{D}$ the distance between checkpoints, $\bar{N}_{min}$ the total simulation run length and P the number of processors [11].

In terms of the combination of results by the global analyser, the local estimates are combined into a global estimate of the mean. This is given as

$$\hat{v} = \frac{\sum_i n_i \hat{v}_i}{\sum_i n_i},$$

where $\hat{v}_i$ is the local point estimate, and n_i the local number of observations, from engine i. The variance of the global estimate is calculated by the global analyser and given as

$$Var[\hat{v}] = \frac{\sum_i n_i^2 Var[\hat{v}_i]}{\left(\sum_i n_i^2\right)^2},$$

where $Var[\hat{v}_i]$ is the variance of the local estimate at engine i.

The resulting global estimates can be assumed to follow a Student t-distribution with $\sum_i d_i$ degrees of freedom. Using this information, a global point estimate with corresponding confidence interval can be constructed.

3. Terminating Simulation

Terminating simulation, alternatively known as finite time-horizon simulation, is simulation that is stopped when a given stopping criteria is met. The analysis methodology used for this type of simulation is *Independent Replications* (IR).

Following the methodology of Independent Replications, we execute different replications using non-overlapping (statistically independent) sequences of pseudo-random numbers. We begin with some minimum, N_{min}, number of replications and then set the next checkpoint to be after the next replication. This is shown in Figure 1.

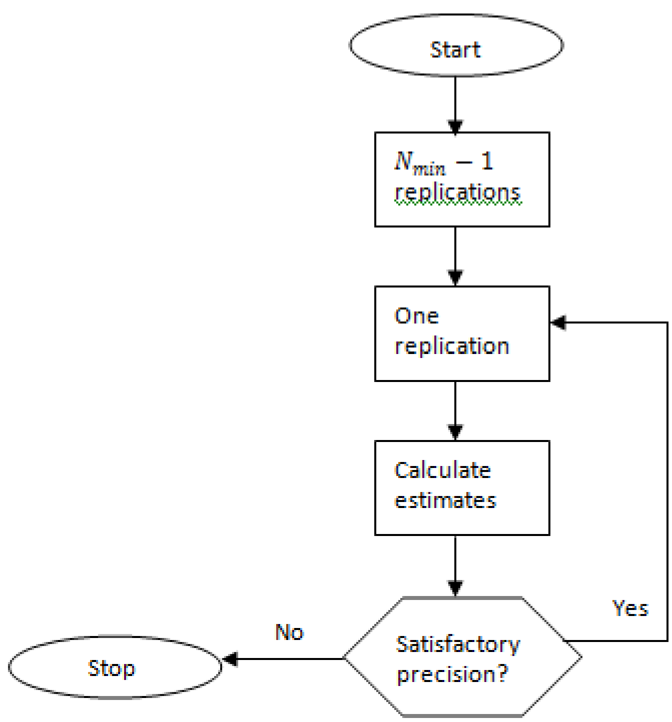

Figure 1 – Sequential terminating simulation based on Independent Replications

If we let $x_1, x_2, \ldots, x_n$ be the simulation output from replication 1, replication 2, ..., replication n, then we see that they represent independent and identically distributed random variables $X_1, X_2, \ldots, X_n$.

3.1. Terminating Simulation Variance

The point estimator and its corresponding confidence interval for terminating simulation is relatively straight forward to obtain. Making use of the property described earlier, that the output from each replication of simulation represents an independent and identically distributed random variable, we can create the following point estimate.

$$\hat{\sigma}^2_{Term} = \frac{1}{n-1}\sum_{i=1}^{n} x_i^2 - \frac{n}{n-1}\bar{x}^2,$$

where

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

is the sample mean of the outputs from each replication.

Using equation 2.4, the variance of $\hat{\sigma}^2_{IR}$ can be estimated as

$$Var[\hat{\sigma}^2_{Term}] = \frac{1}{n}\left(\sum_{i=1}^{n}(x_i - \bar{x})^4 - \frac{n-3}{n-1}\hat{\sigma}^4_{Term}\right)$$

$$= \frac{1}{n^2}\sum_{i=1}^{n} x_i^4 - \frac{4}{n^2}\bar{x}\sum_{i=1}^{n} x_i^3 + \frac{6}{n^2}\bar{x}^2\sum_{i=1}^{n} x_i^2 - \frac{3}{n}\bar{x}^4 - \frac{n-3}{n(n-1)}\hat{\sigma}^4_{IR}.$$

The corresponding confidence interval is calculated as

$$\hat{\sigma}^2_{Term} - \Delta_{Term} < \sigma^2 < \hat{\sigma}^2_{Term} + \Delta_{Term},$$

where

$$\Delta_{Term} = z_{1-\alpha/2}\sqrt{Var[\hat{\sigma}^2_{Term}]}.$$

As there are no general results that exist on the distribution of $S^2(n)$, and we are considering cases where n is large, we assume a normal distribution [2].

4. Steady-State Simulation

Because of the characteristics of steady-state simulation, finding an accurate estimator of variance is much more challenging than that of terminating simulation. Several estimators have been published in [2]. In many ways this work is an extension of previous research, most notably as additional tests are applied on some estimators, as well as greater detail on the tests that have already been applied. The estimators that are analysed here are those that were recommended by Schmidt, as well as those not fully tested.

The three estimators investigated are based on completely different approaches. The first estimator discussed is based on the principle of independent replications, similar to that used in terminating simulation. The second estimator is based on variance as a mean value, and the last estimator based on batch means. Each estimator is discussed further in its corresponding section. Please note that the following equations and algorithms related to each estimator were originally given by Schmidt in [2].

One issue present in steady-state simulation is dealing with the initial transient period where observations don't represent steady-state processes. In the case of simulation of variance, no generally applicable method for initial transient period has been proposed. One method that could be applied after further research has been proposed by Eickhoff in [20]. In this research we have assumed that initial transient is shorter than some fixed number of observations, which is stated in the case of each estimator. The observations that are recorded during the specified fixed period are discarded. Development of an accurate initial transient period of variance has been left for further research.

4.1. Independent Replications

This estimator is based on the work in [2 pp. 23-25]. The principle of independent replications is based on the idea that it is possible to use the sample variance as a point estimator $S^2(n)$, from 2.3, if the observations are uncorrelated. If the observations are unbiased, then $S^2(n)$ can be used as a point estimator its variance and equation 2.4 can be used to construct the corresponding interval.

One way to ensure that the observations used are uncorrelated is to use independent replications of the simulation model. From each replication only a single observation is taken from the steady-state phase of simulation. The disadvantage of this methodology is that many observations are discarded, and the runtime of the simulation is large. This is evident in the mean run length of simulation shown in the appendix in Table 9, Table 10 and Table 11. Note that in these tables, the total number of observations used in a single simulation is the mean run length multiplied by the length of the initial transient period.

The resulting observations from such an independent replication methodology are

$$
\begin{array}{l}
x_{1,1}, x_{1,2}, x_{1,3}, \ldots \text{ first replication,} \\
x_{2,1}, x_{2,2}, x_{2,3}, \ldots \text{ second replication,} \\
\qquad\quad \vdots \\
\qquad\quad \vdots \\
x_{i,1}, x_{i,2}, x_{i,3}, \ldots \ i\text{th replication} \\
\qquad\quad \vdots \\
\qquad\quad \vdots
\end{array}
$$

From this we define the observation

$$y_i = x_{i,n_{o,i}},$$

where $n_{o,i}$ is the truncation point of the ith replication.

As each y_i is obtained from independent replications, they are independent and identically distributed, so we can apply the sample variance $S^2(n)$ as an estimator of σ^2. We call this point estimate $\hat{\sigma}^2_{IR}$. We can use equation 2.4 to construct a confidence interval for $\hat{\sigma}^2_{IR}$. The following equations are then derived

$$\hat{\sigma}_{IR}^2 = \frac{1}{n-1}\sum_{j=1}^{n} y_j^2 - \frac{n}{n-1}\bar{y}^2,$$

where

$$\bar{y} = \frac{1}{n}\sum_{j=1}^{n} y_j$$

is the sample mean of $\{\, y\}$.
Using equation 2.4, the variance of $\hat{\sigma}_{IR}^2$ can be estimated as

$$Var[\hat{\sigma}_{IR}^2] = \frac{1}{n}\left(\sum_{j=1}^{n}\left(y_j - \bar{y}\right)^4 - \frac{n-3}{n-1}\hat{\sigma}_{IR}^4\right)$$

$$= \frac{1}{n^2}\sum_{j=1}^{n} y_j^4 - \frac{4}{n^2}\bar{y}\sum_{j=1}^{n} y_j^3 + \frac{6}{n^2}\bar{y}^2\sum_{j=1}^{n} y_j^2 - \frac{3}{n}\bar{y}^4 - \frac{n-3}{n(n-1)}\hat{\sigma}_{IR}^4.$$

The corresponding confidence interval is calculated as

$$\hat{\sigma}_{IR}^2 - \Delta_{IR} < \sigma^2 < \hat{\sigma}_{IR}^2 + \Delta_{IR},$$

where

$$\Delta_{IR} = z_{1-\alpha/2}\sqrt{Var[\hat{\sigma}_{IR}^2]}.$$

As was discussed earlier (in the terminating simulation section), no general results exist on the distribution of $S^2(n)$, and as we are considering cases where n is large, we assume a normal distribution [2].

It is important to note that the equations used for this estimator have been transformed slightly in order to make them more computationally efficient. We see that in order to implement this estimator, we need only keep track of y_j, y_j^2, y_j^3 and y_j^4.

The algorithm for implementation of $\hat{\sigma}_{IR}^2$is described below.

Implementation Algorithm for Estimator $\hat{\sigma}_{IR}^2$

$n = 0$
$\sum y = 0,\ \sum y^2 = 0, \sum y^3 = 0, \sum y^4 = 0$
Repeat
    $Restart\ Simulation\ Model$
    Repeat
        $x = Get\ Observation$
    Until $Initial\ Transient\ Period\ Over$
    $y = Get\ Observation$
    $n = n + 1$
    $\sum y = \sum y + y$
    $\sum y^2 = \sum y^2 + y^2$
    $\sum y^3 = \sum y^3 + y^3$
    $\sum y^4 = \sum y^4 + y^4$
    $\bar{y} = \sum y / n$
$\hat{\sigma}_{IR}^2 = \sum y^2/(n-1) - n\,\bar{y}^2/(n-1)$
$v = \sum y^4/n^2 - 4\,\bar{y} \sum y^3/n^2 + 6\bar{y} \sum y^2/n^2 - 3\bar{y}^4/n - (\hat{\sigma}_{IR}^2)^2\,(n-3)/n(n-1)$
$\Delta = z_{1-\alpha/2}\sqrt{v}$
Until $RequiredPrecision(\hat{\sigma}_{IR}^2, \Delta)$
Return $\hat{\sigma}_{IR}^2, [\hat{\sigma}_{IR}^2 - \Delta, \hat{\sigma}_{IR}^2 + \Delta]$

4.2. Variance as a Mean Value

This estimator is based on the work in [2 pp. 20-21]. It follows approach to estimation of variance derived from the definition of variance as

$$Var[X] = E[(X - E[X])^2].$$

By keeping track of this parameter as a mean value, we can use existing mean value estimation techniques to calculate the variance. In this paper, the mean of *spectral analysis* is used [12]. This method produces good results in terms of coverage for mean values, as has been shown in [19].

In order to implement this estimator, we transform each observation to

$$y_i = (x_i - \bar{x}(i))^2,$$

where x_i is the ith observation and

$$\bar{x}(i) = \frac{1}{i}\sum_{j=1}^{i} x_i$$

the sample mean of the first i observations.

The estimator can then be defined as the mean value of $y_1, y_2, y_3, \ldots$, that is,

$$\hat{\sigma}_M^2(n) = \frac{1}{n}\sum_{i=1}^{n} y_i.$$

Using spectral analysis as the mean value analysis method, the value $Var[\hat{\sigma}_M^2(n)]$ can be calculated, and hence the lower and upper bounds of the confidence interval,

$$\bar{y}_l(n) = \hat{\sigma}_M^2(n) - t_{d,1-\alpha/2}\sqrt{Var[\hat{\sigma}_M^2(n)]},$$

$$\bar{y}_h(n) = \hat{\sigma}_M^2(n) + t_{d,1-\alpha/2}\sqrt{Var[\hat{\sigma}_M^2(n)]}.$$

We then have the following symmetric confidence interval

$$\bar{y}_l(n) < \sigma^2 < \bar{y}_h(n).$$

The following implementation algorithm for $\hat{\sigma}_M^2$ is given.

Implementation Algorithm for Estimator $\hat{\sigma}_M^2$

$\sum x = 0, n = 0$
Repeat
    $x = Get\ Observation$
    $\sum x = \sum x + x$
    $n = n + 1$
    $y = (x - \sum x/n)^2$
    $SumbitToEstimatorOfMean(y)$
 $\hat{\sigma}_M^2, \Delta = CurrentEstimateOfMean$
Until $RequiredPrecision(\hat{\sigma}_M^2, \Delta)$
Return $\hat{\sigma}_M^2, [\hat{\sigma}_M^2 - \Delta, \hat{\sigma}_M^2 + \Delta]$

4.3. Batch Means

This estimator is based on the work in [2 pp. 32-36]. Batch means is a method of batching observations together and extracting a property from the batch, such as the mean, which should have a lower correlation with other batch means than comparing individual observations from different batches. This technique is described the paper [13], where the methodology for estimating steady-state variance via batch means is discussed. In the paper from Feldman, Deuermeyer and Yang, the possibility of compensating for the bias of sample variance $S^2(n)$ is a main point of interest.

The idea behind calculating variance via batch means, is splitting the variance into two components:

- Local Variance – Mean variance inside batches
- Global Variance – Variance of means of the batches

Taking batches with size b, we calculate the sample mean and sample variance as

$$\bar{X}_j = \frac{1}{n}\sum_{k=1}^{n} X_{j,k},$$

$$S_j^2 = \frac{1}{n}\sum_{k=1}^{n} (X_{j,k} - \bar{X}_j)^2,$$

where each batch j consists of $X_{j,1}, X_{j,2}, \dots, X_{j,n}$ observation.

Using these values, the sample mean $\bar{\bar{X}}$ and sample variance $S_{\bar{X}}^2$of the batch means can be calculated as

$$\bar{\bar{X}} = \frac{1}{b}\sum_{j=1}^{b} \bar{X}_j,$$

$$S_{\bar{X}}^2 = \frac{1}{b-1}\sum_{j=1}^{b} (\bar{X}_j - \bar{\bar{X}})^2.$$

The mean batch variance $\bar{V}$is then calculated as

$$\bar{V} = \frac{1}{b}\sum_{j=1}^{b} S_j^2,$$

which leads to the point estimate

$$\hat{\sigma}^2 = \bar{V} + S_{\bar{X}}^2.$$

In order to show that $\hat{\sigma}^2$ is an accurate estimator of σ^2, we show that

$$\begin{aligned} E[\hat{\sigma}^2] &= E[\bar{V}] + E[S_{\bar{X}}^2] \\ &= E\left[\tfrac{m-1}{m} S^2(m)\right] + Var[\bar{X}] \\ &= \tfrac{m-1}{m}\sigma^2 - \tfrac{2\sigma^2}{m}\xi_m + \tfrac{1}{m}\sigma^2 + \tfrac{2\sigma^2}{m}\xi_m \\ &= \sigma^2, \end{aligned}$$

where $\xi_m = \sum_{j=1}^{m-1}(1 - j/m)\rho_j$ is the autocorrelation term. Hence we have an unbiased estimator $\hat{\sigma}^2$.

In order to obtain a confidence interval for the point estimate $\hat{\sigma}^2$, we look at the variable Y_j which has a sample mean equal to $\hat{\sigma}^2$. That is,

$$Y_j = S_j^2 + \frac{b}{b-1}\left(\bar{X}_j - \bar{\bar{X}}\right)^2.$$

The sample variance of Y_j can then be obtained, which is used to construct the confidence interval for $\hat{\sigma}^2$. This is calculated as

$$S_y^2 = \frac{1}{b-1}\sum_{j=1}^{b}\left(Y_j - \hat{\sigma}^2\right)^2.$$

In order turn the derived equations from above into ones suitable for sequential estimation, slight transformations are made. These transformations are incorporated into the construction of estimator $\hat{\sigma}_{BM}^2$, which is described below.

$$\begin{aligned}
\hat{\sigma}_{BM}^2 &= \frac{1}{b}\sum_{j=1}^{b} y_j \\
&= \frac{1}{b}\sum_{j=1}^{b} s_j^2 + \frac{1}{b-1}\sum_{j=1}^{b}\left(x_j - \bar{\bar{x}}\right)^2 \\
&= \frac{1}{b}\sum_{j=1}^{b} s_j^2 + \frac{1}{b-1}\sum_{j=1}^{b} \bar{x}_j^2 - \frac{b}{b-1}\bar{\bar{x}}^2
\end{aligned}$$

The confidence interval for $\hat{\sigma}_{BM}$is then constructed using the sample variance of y_j, that is,

$$\begin{aligned}
S_y^2 &= \frac{1}{b-1}\sum_{j=1}^{b}\left(y_j - \hat{\sigma}_{BM}^2\right)^2 \\
&= \frac{1}{b-1}\sum_{j=1}^{b}\left(y_j^2 - 2y_j\hat{\sigma}_{BM}^2 + \hat{\sigma}_{BM}^4\right) \\
&= \frac{1}{b-1}\sum_{j=1}^{b} y_j^2 - \frac{b}{b-1}\hat{\sigma}_{BM}^4,
\end{aligned}$$

where

$$\begin{aligned}
\sum_{j=1}^{b} y_j^2 &= \sum_{j=1}^{b}\left(s_j^2 + \frac{b}{b-1}\left(\bar{x}_j - \bar{\bar{x}}\right)^2\right)^2 \\
&= \sum_{j=1}^{b} s_j^4 + \frac{2b}{b-1}\left(\sum_{j=1}^{b} s_j^2\bar{x}_j^2 - 2\bar{\bar{x}}\sum_{j=1}^{b} s_j^2\bar{x}_j + \bar{\bar{X}}^2\sum_{j=1}^{b} s_j^2\right) + \left(\frac{b}{b-1}\right)^2\sum_{j=1}^{b}\left(\bar{x}_j - \bar{\bar{x}}\right)^4,
\end{aligned}$$

and

$$\sum_{j=1}^{b}\left(\bar{x}_j - \bar{\bar{x}}\right)^4 = \sum_{j=1}^{b} {\bar{x}_j}^4 - 4\bar{\bar{x}}\sum_{j=1}^{b} {\bar{x}_j}^3 + 6\bar{\bar{x}}^2\sum_{j=1}^{b} {\bar{x}_j}^2 - 3b\bar{\bar{x}}^4.$$

We can then create the corresponding confidence interval

$$\hat{\sigma}^2_{BM} - \Delta_{BM} < \sigma^2 < \hat{\sigma}^2_{BM} + \Delta_{BM}$$

where

$$\Delta_{BM} = t_{b-1,1-\alpha/2}\sqrt{s_y^2/b}$$

and $t_{b-1,1-\alpha/2}$ is the $1-\alpha/2$ quantile of the Student t-distribution with $b-1$ degrees of freedom.

Using these equations designed for sequential use, only certain variables need to be stored and updated in each iteration of the algorithm. These variables are s_j^2, s_j^4, $\bar{x}_j$, $\bar{x}_j{}^2$, $\bar{x}_j{}^3$, $\bar{x}_j{}^4$, $s_j^2\bar{x}_j$ and $s_j^2\bar{x}_j^2$.

The algorithm can be implemented as follows.

Implementation Algorithm for Estimator $\hat{\sigma}^2_{BM}$

$m = InputBatchSize$
$\sum\bar{x} = 0,\ \sum\bar{x}^2 = 0, \sum\bar{x}^3 = 0, \sum\bar{x}^4 = 0$
$\sum s^2 = 0, \sum s^4 = 0, \sum s^2\bar{x} = 0, \sum s^2\bar{x}^2 = 0$
$b = 0$
Repeat
  $\sum x = 0, \sum x^2 = 0$
  For $i = 1$ to m **do** (Collect batch of observations)
    $x = GetObservation$
    $\sum x = \sum x + x$
    $\sum x^2 = \sum x^2 + x^2$
  End for
  $b = b + 1$
  $\bar{x} = \sum x/m$
  $s^2 = \sum x^2 - (\sum x)^2$
  $\sum\bar{x} = \sum\bar{x} + \bar{x}$
  $\sum\bar{x}^2 = \sum\bar{x}^2 + \bar{x}^2$
  $\sum\bar{x}^3 = \sum\bar{x}^3 + \bar{x}^3$
  $\sum\bar{x}^4 = \sum\bar{x}^4 + \bar{x}^4$
  $\sum s^2 = \sum s^2 + s^2$
  $\sum s^4 = \sum s^4 + (s^2)^2$
  $\sum s^2\bar{x} = \sum s^2\bar{x} + s^2\cdot\bar{x}$
  $\sum s^2\bar{x}^2 = \sum s^2\bar{x}^2 + s^2\cdot\bar{x}^2$
  $v = \sum\bar{x}/b$
  $\hat{\sigma}^2_{BM} = \sum s^2/b + \sum\bar{x}^2/(b-1) - bv^2/(b-1)$
  $\sum y^2 = \sum s^4 + \frac{2b}{b-1}\left(\sum s^2\bar{x}^2 - 2v\sum s^2\bar{x} + v^2\sum s^2\right)$
    $+\left(\frac{c}{c-1}\right)^2\left(\sum\bar{x}^4 - 4v\sum\bar{x}^3 + 6v^2\sum\bar{x}^2 - 3bv^4\right)$
  $\Delta = t_{b-1,1-\alpha/2}\sqrt{S_Y^2/b}$
Until $RequiredPrecision(\hat{\sigma}^2_{BM}, \Delta)$
Return $\hat{\sigma}^2_{BM}, [\hat{\sigma}^2_{BM} - \Delta, \hat{\sigma}^2_{BM} + \Delta]$

4.4. Evaluation of Estimators

All three steady-state variance estimators have been tested to a certain degree by Schmidt et al [2]. However, on occasion there were places were more testing could have been performed. This chapter explains how the estimators were evaluated and the rationale behind such investigation.

4.4.1. Reference Models

There are three different queuing models considered in this paper, although one model is used twice with different parameters. The queues considered are the single server queues M/M/1, M/E_2/1 and M/H_2/1. The coefficient of variation for the service times used are $C_{st} = 1$ for the M/M/1 queue, $C_{st} < 1$ for the M/E_2/1 queue and $C_{st} = 5, 50$ for the M/H_2/1 queue. The reason these queues are considered is that it allows independent verification of results presented for the estimators in [2]. It also allows expansion on previous work with these estimators and queuing models, by allowing the testing of higher loads. Use of the M/H_2/1 queue with $C_{st} = 50$ allows investigation of a queue with heavy-tailed distribution. The property of each queue investigated is the waiting time.

The Pollaczek-Khintchine transform formula $W_q(s)$ for the waiting time distribution is used to obtain the variance of the waiting time [2], [14]. The nth moment of the waiting time distribution can be found by differentiating $W_q(s)$ n times and evaluating at $s = 0$. Therefore we can calculate the mean and variance as

$$v = \lim_{s \to 0} -\frac{dW_q(s)}{ds},$$

and

$$\sigma^2 = \lim_{s \to 0} \frac{d^2 W_q(s)}{ds} - v^2.$$

For each of the queues investigated (M/M/1, M/E_2/1 and M/H_2/1), the equations presented are given in [2 pp. 40-43], with some corrections and alterations made.

4.4.1.1. M/M/1 Queue

The M/M/1 queue is a single-sever queue with arrival rate λ from a Poisson process and service time μ from an exponential distribution. The load of the queue is calculated as $\rho = \lambda/\mu$.

As was discussed earlier, the variance of the waiting time can be derived from the Pollaczek-Khintchine transform formula $W_q(s)$ for the waiting time distribution. For the M/M/1 queue we have

$$W_q(s) = \frac{\left(1 - \frac{\lambda}{\mu}\right) s}{s - \lambda\left(1 - \frac{\mu}{\mu + s}\right)}.$$

From this we can calculate the mean, v, and variance, σ^2, of the waiting time as

$$v = \frac{\lambda}{\mu(\mu - \lambda)}$$

and

$$\sigma^2 = \frac{\lambda(2\mu - \lambda)}{\mu^2(\mu - \lambda)^2}.$$

These formulas are evaluated to produce the values in Table 1.

System Load ρ	Arrival Rate λ	Service Rate μ	Mean Waiting Time v	Variance of Waiting Time σ^2
0.1	0.1	1	0.111	0.235
0.2	0.2	1	0.25	0.563
0.3	0.3	1	0.429	1.041
0.4	0.4	1	0.667	1.778
0.5	0.5	1	1	3
0.6	0.6	1	1.5	5.25
0.7	0.7	1	2.333	10.111
0.8	0.8	1	4	24
0.9	0.9	1	9	99
0.95	0.95	1	19	399

Table 1 – Waiting time distribution values for M/M/1 queue

4.4.1.2. $M/E_2/1$ Queue

The $M/E_2/1$ queue has a an arrival rate λ, which follows a Poisson process. The service rate, μ, is given by an Erlang-2 distribution (a distribution with two equal stages of service). The resulting load is given by $\rho = 2\lambda/\mu$.

Again the Pollaczek-Khintchine transform formula for waiting time distribution is used to derive the mean and variance of waiting time. For the $M/E_2/1$ queue, we use

$$W_q(s) = \frac{\left(1 - \frac{2\lambda}{\mu}\right)s}{s - \lambda\left(1 - \frac{\mu^2}{(\mu + s)^2}\right)},$$

from which we calculate the mean, v, and variance, σ^2, as

$$v = \frac{3\lambda}{\mu(-2\lambda)}$$

and

$$\sigma^2 = \frac{\lambda(8\mu - 7\lambda)}{\mu^2(\mu - 2\lambda)^2}.$$

These formulas are evaluated to produce the values in Table 2.

System Load ρ	Arrival Rate λ	Service Rate μ	Mean Waiting Time v	Variance of Waiting Time σ^2
0.1	0.1	2	0.083	0.118
0.2	0.2	2	0.188	0.285
0.3	0.3	2	0.321	0.532
0.4	0.4	2	0.5	0.917
0.5	0.5	2	0.75	1.563
0.6	0.6	2	1.125	2.766
0.7	0.7	2	1.75	5.396
0.8	0.8	2	3	13
0.9	0.9	2	7.75	54.563
0.95	0.95	2	14.25	222.0625

Table 2 - Waiting time distribution values for M/E_2/1 queue

4.4.1.3. M/H_2/1 Queue

The arrival rate, λ, for the M/H2/1 queue is from a Poisson process, while the service rate, μ, is governed by a hyper-exponential distribution. The service time received by each customer is modelled by an exponential distribution with rate μ_1 with a probability p_1 and μ_2 with a probability p_2. The corresponding Pollaczek-Khintchine transform formula for the waiting time distribution is

$$W_q(s) = \frac{\left(1 - \lambda\left(\frac{p_1}{\mu_1} + \frac{1-p_1}{\mu_2}\right)\right)s}{s - \lambda\left(1 - \frac{p_1\mu_1}{\mu_1 + s} - \frac{(1-p_1)\mu_2}{\mu_2 + s}\right)},$$

which yields the mean, , v, and variance, σ^2, as

$$v = \frac{\lambda(\mu_1^2 - \mu_1^2 p + \mu_2^2 p)}{\mu_1\mu_2(\mu_1\mu_2 - \lambda\mu_2 p - \lambda\mu_1 + \lambda\mu_1 p)}$$

and

$$\sigma^2 = \frac{\lambda}{\mu_1^2\mu_2^2(\mu_1\mu_2 - \lambda\mu_2 p - \lambda\mu_1 + \lambda\mu_1 p)^2}(2\mu_1^3\mu_2\lambda p - 2\mu_1^3\mu_2\lambda p^2 - 2\mu_1\mu_2^2\lambda p^2 + 2\mu_1\mu_2^3\lambda p + 2\mu_1^4\mu_2 p - 2\mu_1\mu_2^4 p \\ - 2\mu_1^4\mu_2 - 2\mu_1^2\mu_2^2\lambda p + 2\mu_1^2\mu_2^2\lambda p^2 + \mu_1^4\lambda - 2\mu_1^4\lambda p + \mu_1^4\lambda p^2 + \mu_2^4\lambda p).$$

To work out the values of parameters μ_1, μ_2, p_1 and p_2, the balanced means approached is used and the parameters then derived [18]. This means that we use the normalisation

$$\frac{p_1}{\mu_1} = \frac{p_2}{\mu_2}.$$

The parameters are then given by

$$p_1 = \frac{1}{2}\left(1 + \sqrt{\frac{c_x^2 - 1}{c_x^2 + 1}}\right), p_2 = 1 - p_1,$$

$$\mu_1 = \frac{2p_1}{E[X]}, \mu_2 = \frac{2p_2}{E[X]}.$$

In this paper we investigate the performance of the estimators on two versions of the M/H_2/1 queue. In this first version we set $E[X] = 1$ and $c_x^2 = 5$. This gives the parameters

$$p_1 = 0.9082, p_2 = 0.0918$$
$$\mu_1 = 1.816,\ \mu_2 = 0.184.$$

In the second version we set $E[X] = 1$ and $c_x^2 = 50$. This gives the parameters

$$p_1 = 0.9901, p_2 = 0.0099$$
$$\mu_1 = 1.9802,\ \mu_2 = 0.0198$$

The evaluated values for each version of the queue are shown in Table 3 and Table 4.

System Load ρ	**Arrival Rate λ**	**Service Parameters**			**Mean Waiting Time v**	**Variance of Waiting Time σ^2**
		p_1	μ_1	μ_2		
0.1	0.1	0.9082	1.816	0.184	0.333	3.444
0.2	0.2	0.9082	1.816	0.184	0.75	8.063
0.3	0.3	0.9082	1.816	0.184	1.286	14.51
0.4	0.4	0.9082	1.816	0.184	2	24
0.5	0.5	0.9082	1.816	0.184	3	39
0.6	0.6	0.9082	1.816	0.184	4.5	65.25
0.7	0.7	0.9082	1.816	0.184	7	119
0.8	0.8	0.9082	1.816	0.184	12	264
0.9	0.9	0.9082	1.816	0.184	27	999
0.95	0.95	0.9082	1.816	0.184	55.717	3819.130

Table 3 - Waiting time distribution values for M/H_2/1 queue with $c_x^2 = 5$

System Load ρ	Arrival Rate λ	Service Parameters p_1	μ_1	μ_2	Mean Waiting Time v	Variance of Waiting Time σ^2
0.1	0.1	0.9901	1.9802	0.0198	2.8334	291.363
0.2	0.2	0.9901	1.9802	0.0198	6.3750	678.145
0.3	0.3	0.9901	1.9802	0.0198	10.9286	1212.3
0.4	0.4	0.9901	1.9802	0.0198	17.0001	1989.01
0.5	0.5	0.9901	1.9802	0.0198	25.5001	3200.28
0.6	0.6	0.9901	1.9802	0.0198	38.2502	5288.11
0.7	0.7	0.9901	1.9802	0.0198	59.5004	9490.35
0.8	0.8	0.9901	1.9802	0.0198	102.001	20604.3
0.9	0.9	0.9901	1.9802	0.0198	229.503	75622.1
0.95	0.95	0.9901	1.9802	0.0198	484.512	283203

Table 4 - Waiting time distribution values for M/H_2/1 queue with $c_x^2 = 50$

4.4.2. Independent Replications Evaluation

In the report by Schmidt, estimator $\hat{\sigma}_{IR}^2$ was only tested on the M/M/1 queue [2]. This was strange as all other estimators were also tested on M/E_2/1 and M/H_2/1 queues. In this paper we perform the additional tests of $\hat{\sigma}_{IR}^2$ on the M/E_2/1 and M/H_2/1 queues. The method of Independent Replications could be considered slow and wasteful due to the time it takes to converge and the number of observations it discards. However, it is still important to demonstrate that it provides good results in terms of coverage on all three queuing systems investigated.

4.4.3. Variance as a Mean Value Evaluation

The estimator $\hat{\sigma}_M^2$ was tested on all three queuing models investigated in work by Schmidt [2]. One purpose of investigation here is to confirm the results already produced, as this estimator was a good candidate for implementation into Akaroa2. This results in the testing of the estimator on the M/M/1, M/E_2/1 and M/H_2/1 (with $c_x^2 = 5$) queue. Another aspect of investigation involves testing on queuing models with higher loads and observing the performance. This is important, as estimators often produce results with a lower coverage as the load of a queuing system increases. Finally, the last property investigated is the performance of the estimator on an additional queuing model, M/H_2/1 with a co-efficient of variation of 50. This performance of the estimator on this queuing model is important as it has a much more heavy tailed distribution than the other queuing models tested. This then represents a wider set of possible processes that the estimator has been tested on.

4.4.4. Batch Means

The estimator $\hat{\sigma}_{BM}^2$ is investigated with a slightly different objective in mind to the other estimators. Previous work has been done investigating the coverage of $\hat{\sigma}_{BM}^2$ with specific batch sizes, but less work has been carried out on investigating how the batch size affects the coverage of a given queuing model. The concept of varying batch sizes is investigated in this research, with the objective of determining an optimal batch size for a given queuing model and load in mind. We apply the term ”optimal” somewhat loosely in this context, by which we mean a batch size that provides satisfactory coverage and uses the fewest number of observations. The M/M/1, $M/E_2/1$ and $M/H_2/1$ queues are also investigated to confirm results from other studies. However, the research goes slightly further in testing higher loads on these queuing models. Similarly to the $\hat{\sigma}_M^2$ estimator, $\hat{\sigma}_{BM}^2$ is also tested on a $M/H_2/1$ queue with a large co-efficient of variation to analyse the performance of the estimator on heavy tailed distributions.

The batch sizes used as starting points for the M/M/1 queue were calculated such that successive batch means will have a correlation of less than 0.01. They were calculated from the results in Law and Daley [15] – [16] via [18]. Due to the nature of deriving batch sizes for given queuing systems, the process can be time exhaustive. For this reason, the other queues have similar batch sizes are taken as starting points, but are increased or decreased depending on the expected correlation. As the coverage is being investigated while the batch sizes are halved, the starting point's only requirement is that it is large enough to guarantee sufficient coverage. Although, even if the starting batch size is too small, it can easily be doubled and re-tested. The starting batch sizes for the M/M/1 queue can be seen in Table 5.

Load	Batch Size	Correlation between successive batch means
0.1	30	0.0085
0.2	60	0.0093
0.3	100	0.0099
0.4	170	0.0097
0.5	300	0.0093
0.6	550	0.0092
0.7	1100	0.0093
0.8	2600	0.0099
0.9	11000	0.0099

Table 5 – Starting Batch Sizes for the M/M/1 Queue

5. Results

In this section the results of coverage analysis from the estimators of steady-state variance is presented. All coverage analysis results are calculated at the 95% confidence level with a relative error of 5%. A brief discussion is also given relating to specific performance aspects of each estimator.

5.1. Independent Replications

The coverage analysis results from the Independent Replications ($\hat{\sigma}_{IR}^2$) estimator are shown in Figure 2, Figure 3 and Figure 4. In each of these figures, a fixed initial transient period was used. The lengths of the initial transient periods differ between queues, as the lengths used for the M/M/1 queue were chosen to be equal to that presented in [2]. This allows the work in [2] to be independently verified. The lengths of the initial transient period assumed for the two other queues were generous, so that the initial transient period would be well over before the collection of observations. For more information on the lengths of initial transient, see Table 9, Table 10 and Table 11 in the appendix.

The results shown in Figure 2 confirm the results given by Schmidt in [2]. The other two figures show results from queues not tested in [2]. They show that the estimator provides satisfactory coverage for all loads investigated. The main issue with this estimator is its inefficiency. Because of the number of observations it discards, it cannot be recommended for practical use. It is also reliant on the detection of the initial transient period.

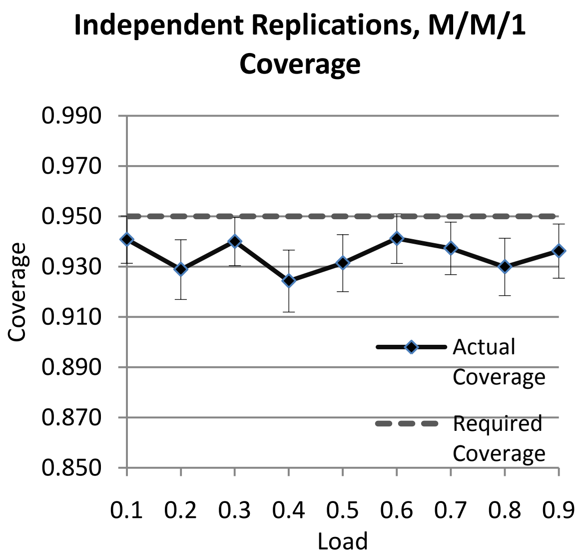

Figure 2 – Coverage of $\hat{\sigma}^2_{IR}$ estimator on M/M/1 queue

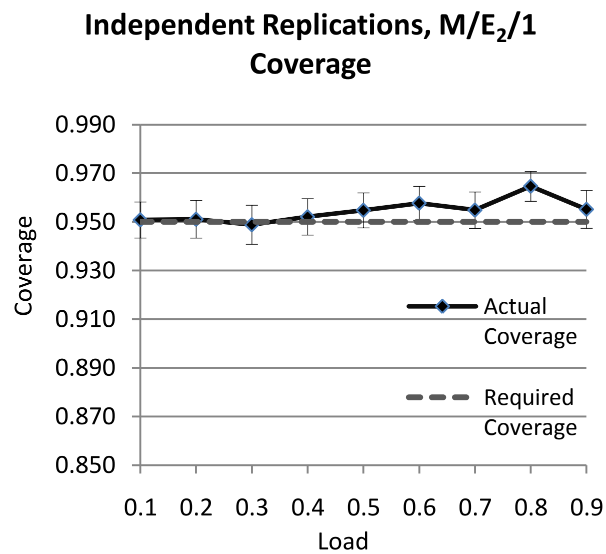

Figure 3 - Coverage of $\hat{\sigma}^2_{IR}$ estimator on M/E$_2$/1 queue

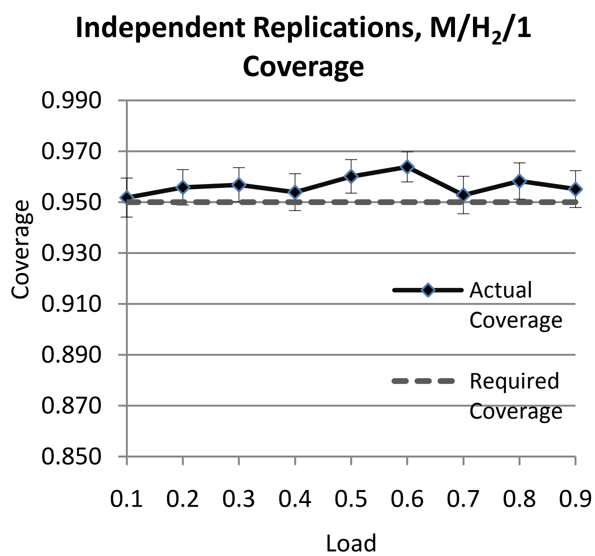

Figure 4 - Coverage of $\hat{\sigma}^2_{IR}$ estimator on M/H$_2$/1 queue

5.2. Variance as a Mean Value

The coverage analysis results from the Variance as a Mean Value estimator ($\hat{\sigma}^2_M$) are shown in Figure 6, Figure 7, Figure 8 and Figure 8. In Figure 6 and Figure 7 a fixed initial transient period of 10,000 observations is used, while Figure 8 and Figure 8 a fixed initial transient period of 20,000 observations is used. This estimator was tested by Schmidt on M/M/1, M/E_2/1 and M/H_2/1 queues with loads 0.1 – 0.9. The results presented here, namely Figure 6, Figure 7 and Figure 7 concur with the results produced in [2]. Thus Figure 6, Figure 7 and Figure 7 are an independent verification of results from [2]. An extension here was made to test higher loads on the M/M/1, M/E_2/1 and M/H_2/1 queues, and also test on an M/H_2/1 queue with a large coefficient of variation. Figure 5 - Figure 8 show that $\hat{\sigma}_M^2$ performs well in terms of coverage, for all loads, apart from some sporadic cases in Figure 8. This may be due to the high coefficient of variation in the M/H_2/1 queue, although as the estimator performs well on the majority of loads, the overall coverage is acceptable.

Another feature of this estimator, apart from good coverage, is how easy is to implement as it makes use of existing mean value estimation mechanisms. In terms of the run time, it is similar to the Batch Means method, with no real separation between them determined.

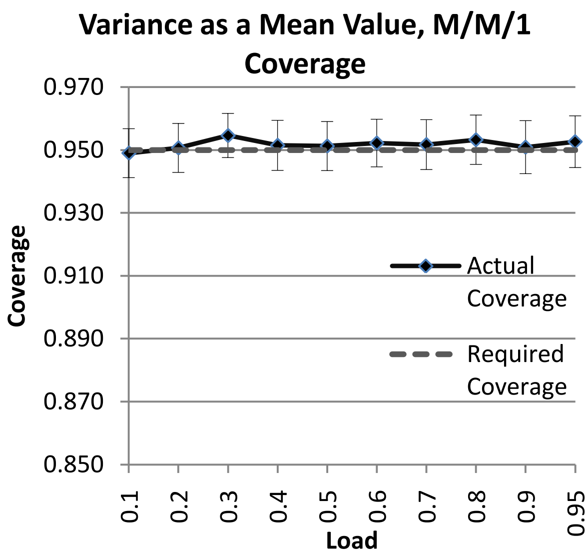

Figure 5 – Coverage of $\hat{\sigma}_M^2$ estimator on M/M/1 queue

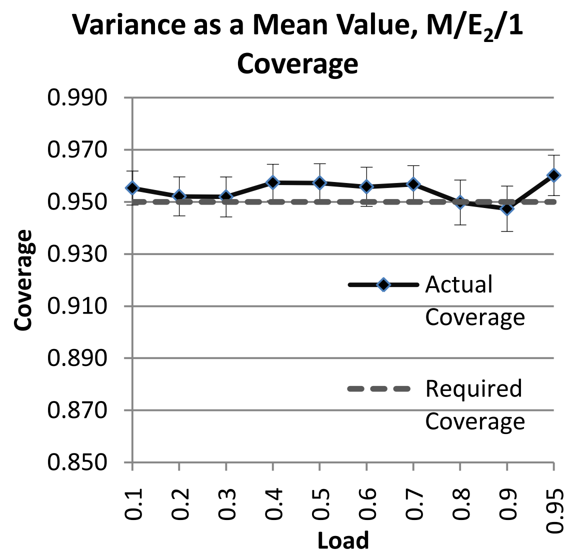

Figure 6 – Coverage of the $\hat{\sigma}_M^2$ estimator on the M/E_2/1 queue

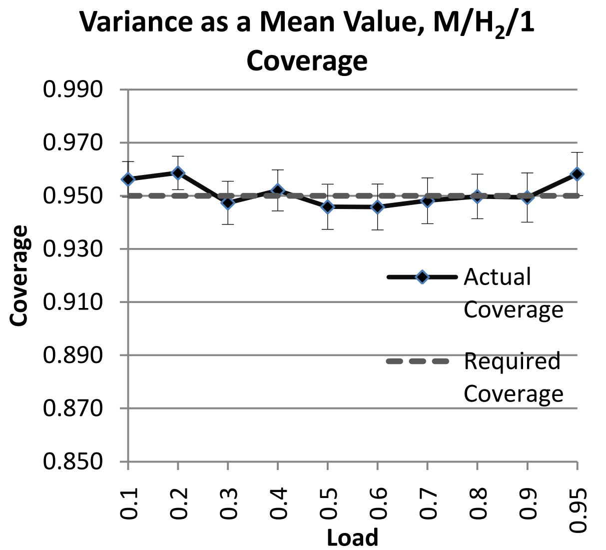

Figure 7 - Coverage of $\hat{\sigma}_M^2$ estimator on M/H$_2$/1 queue, with $c_x^2 = 5$

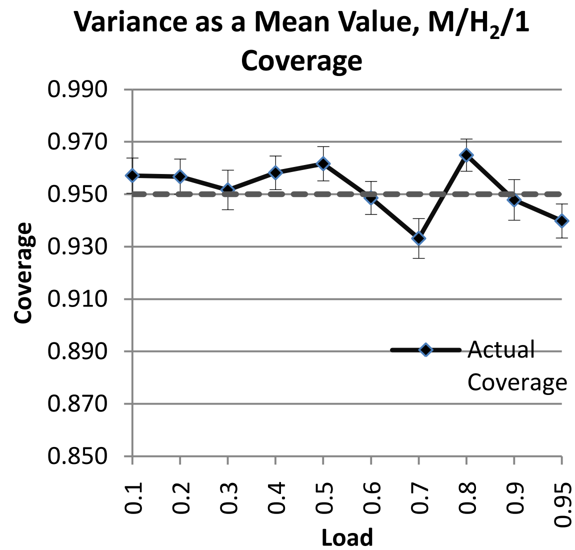

Figure 8 - Coverage of $\hat{\sigma}_M^2$ estimator on M/H$_2$/1 queue, with $c_x^2 = 50$

5.3. Batch Means

In this section, results are presented showing the coverage of the Batch Means ($\hat{\sigma}_{BM}^2$) estimator on various queues. Results from this estimator were presented in [2], but not all loads on queues were investigated. In the results here, all loads from 0.1 – 0.9 are investigated, as well as an additional queue, M/H$_2$/1 with a large coefficient of variation.

The effect that varying batch sizes have on the coverage is shown. Following this, the results show that the estimator performs well for every load and queue investigated, if the correct batch size is known. Optimal batch sizes for each queue investigated are given.

The results confirm those given in [2] and also show that this estimator provides good coverage for queues with larger coefficients of variation, if an appropriate batch size is used.

5.3.1. M/M/1 Queue

The coverage of the Batch Means ($\hat{\sigma}_{BM}^2$) estimator on the M/M/1 queue with varying loads and batch sizes is shown in Figure 9 through to Figure 17. In all of these cases, an initial transient period of 10,000 observations is used. In each of these figures, acceptable coverage can be found if the batch size is large enough. This confirms that $\hat{\sigma}_{BM}^2$ is an accurate estimator for an M/M/1 queue.

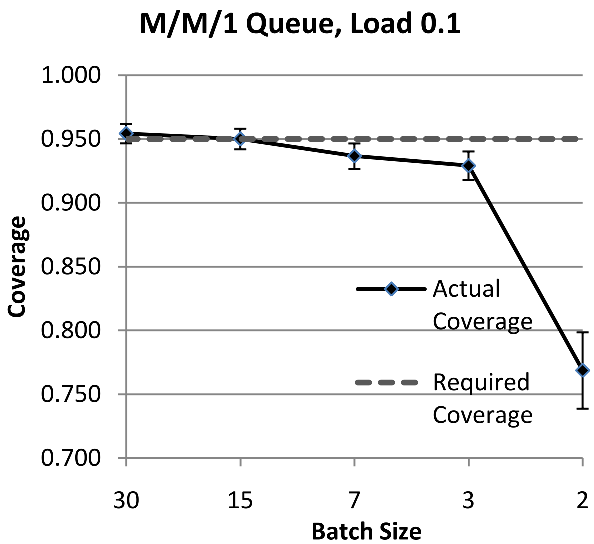

Figure 9 – Coverage of $\hat{\sigma}^2_{BM}$ with varying batch sizes on M/M/1 queue with Load 0.1

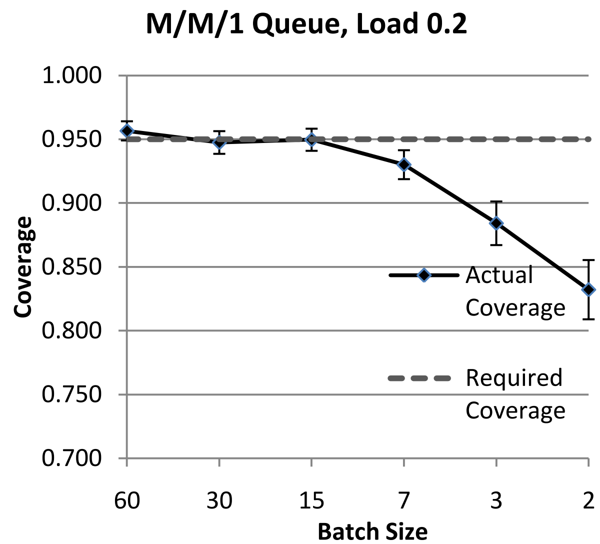

Figure 10 - Coverage of $\hat{\sigma}^2_{BM}$ with varying batch sizes on M/M/1 queue with Load 0.2

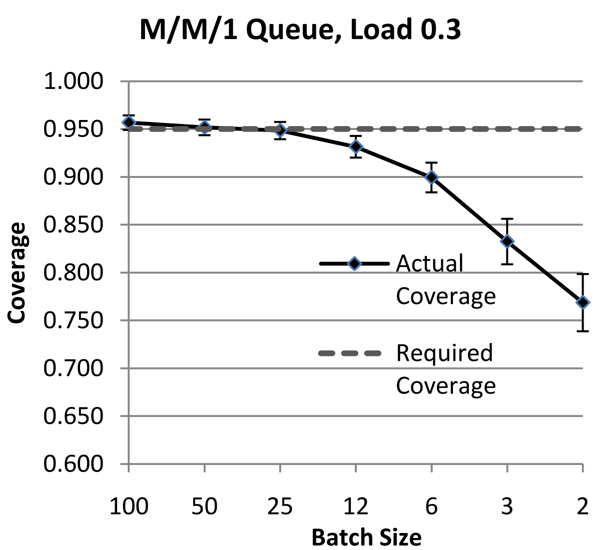

Figure 11 - Coverage of $\hat{\sigma}^2_{BM}$ with varying batch sizes on M/M/1 queue with Load 0.3

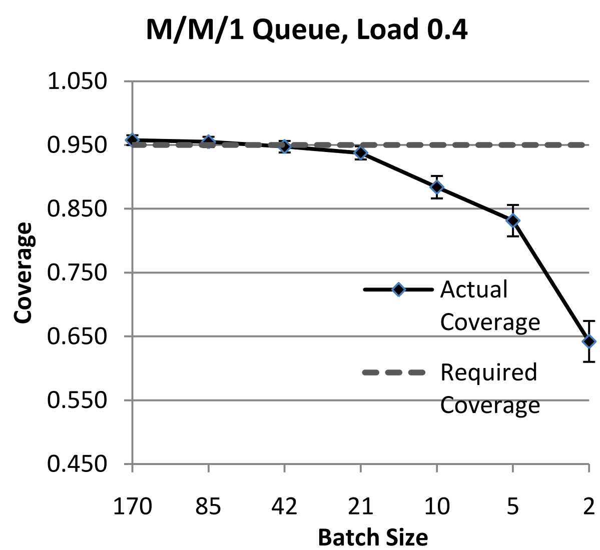

Figure 12 - Coverage of $\hat{\sigma}^2_{BM}$ with varying batch sizes on M/M/1 queue with Load 0.4

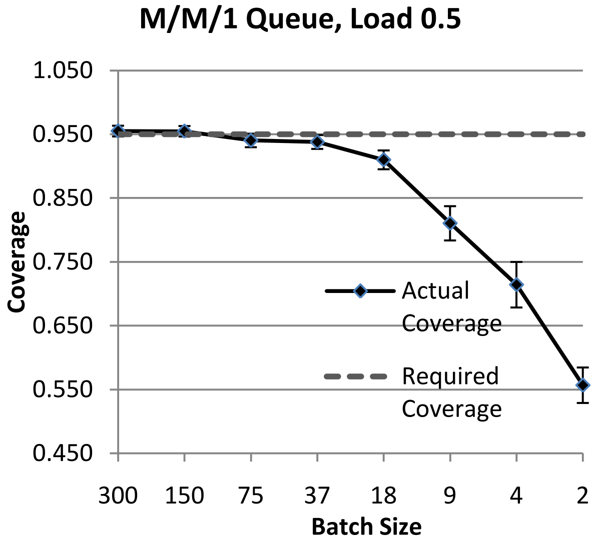

Figure 13 - Coverage of $\hat{\sigma}^2_{BM}$ with varying batch sizes on M/M/1 queue with Load 0.5

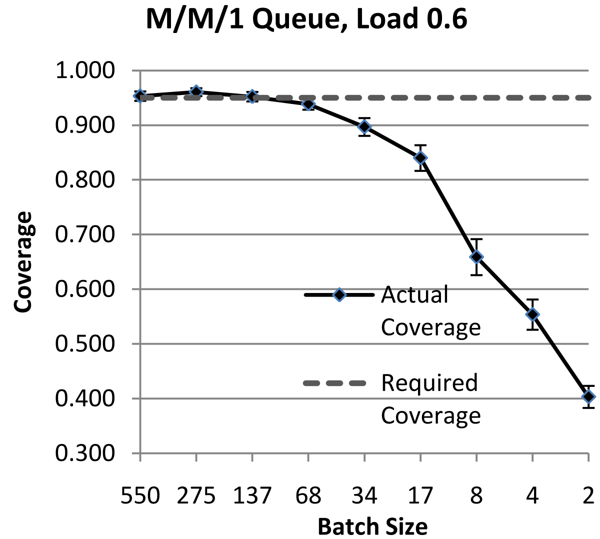

Figure 14 - Coverage of $\hat{\sigma}^2_{BM}$ with varying batch sizes on M/M/1 queue with Load 0.6

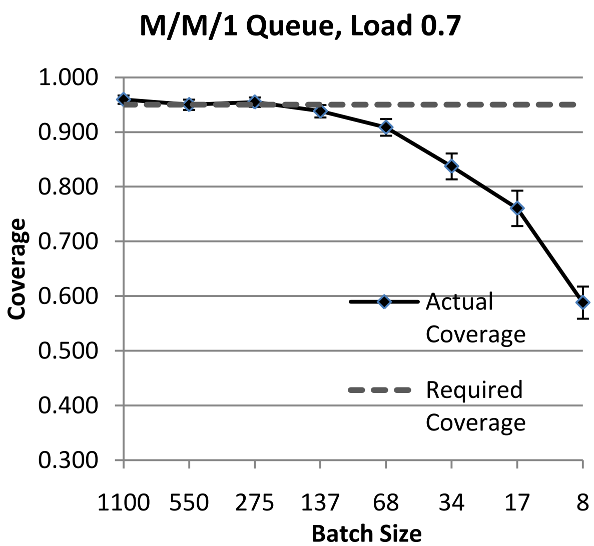

Figure 15 - Coverage of $\hat{\sigma}^2_{BM}$ with varying batch sizes on M/M/1 queue with Load 0.7

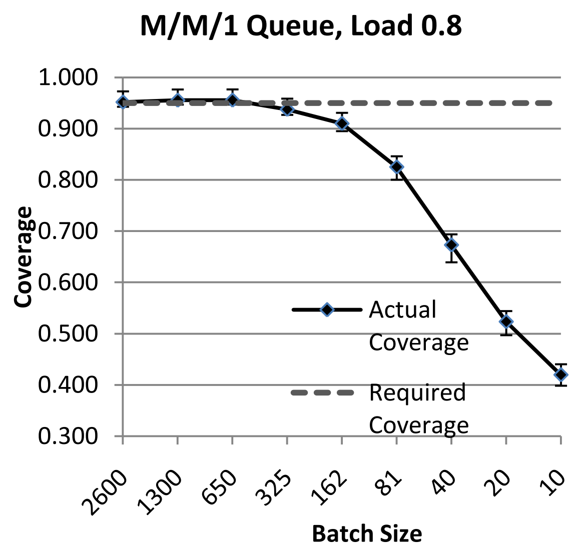

Figure 16 - Coverage of $\hat{\sigma}^2_{BM}$ with varying batch sizes on M/M/1 queue with Load 0.8

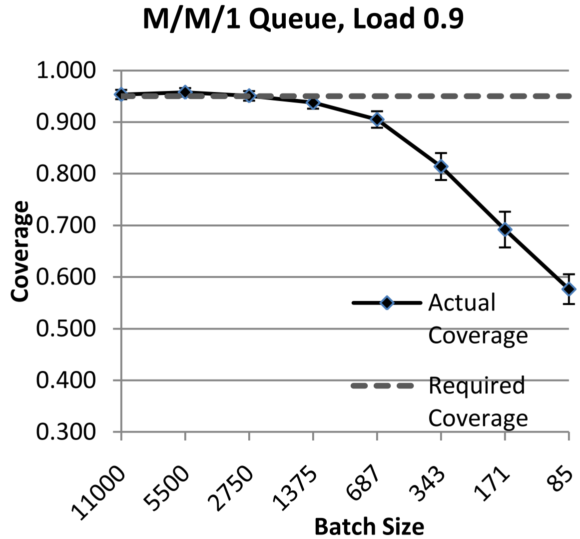

Figure 17 - Coverage of $\hat{\sigma}^2_{BM}$ with varying batch sizes on M/M/1 queue with Load 0.9

From these results we see that for most loads, the coverage is still adequate even after the batch size has been halved three to four times. The resulting "optimal" batch sizes for an M/M/1 queue are shown in Table 6.

Load	Batch Size
0.1	7
0.2	15
0.3	25
0.4	42
0.5	72
0.6	137
0.7	275
0.8	325
0.9	1375

Table 6 – Approximate Optimal Batch Sizes for Variance of Waiting Time of M/M/1 queue

5.3.2. M/E_2/1 Queue

The coverage of the Batch Means ($\hat{\sigma}^2_{BM}$) estimator on the M/E_2/1 queue with varying loads and batch sizes is shown in Figure 18 through to Figure 26. In all of these cases, an initial transient period of 10,000 observations is used. Once again, good coverage is produced in each figure, if the batch size is large enough. The $\hat{\sigma}^2_{BM}$ estimator is therefore a good estimator for the M/E_2/1.

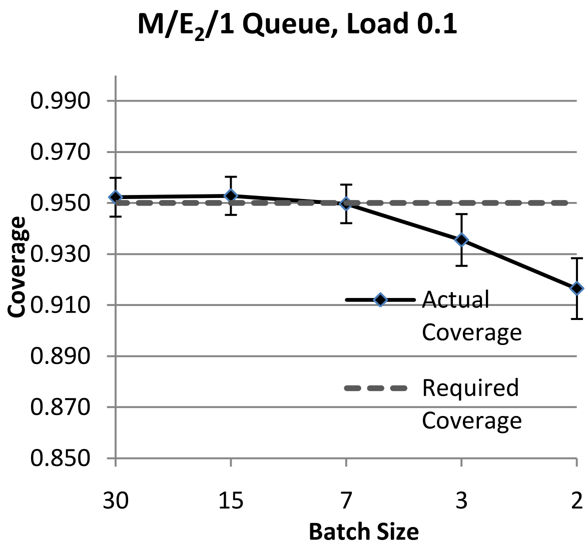

Figure 18 - Coverage of $\hat{\sigma}^2_{BM}$ with varying batch sizes on M/E$_2$/1 queue with Load 0.1

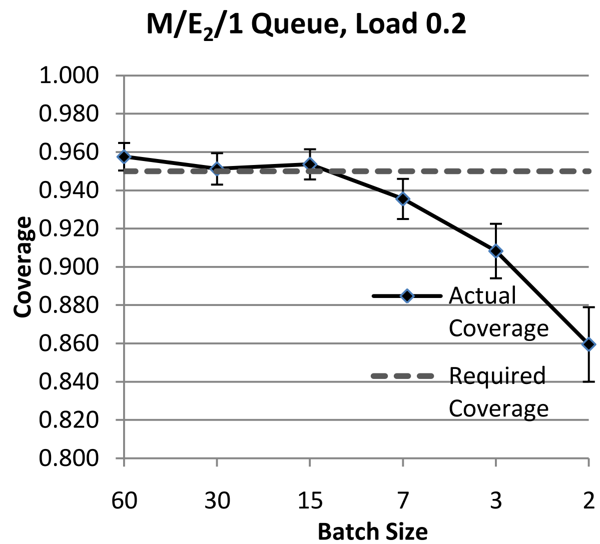

Figure 19 - Coverage of $\hat{\sigma}^2_{BM}$ with varying batch sizes on M/E$_2$/1 queue with Load 0.2

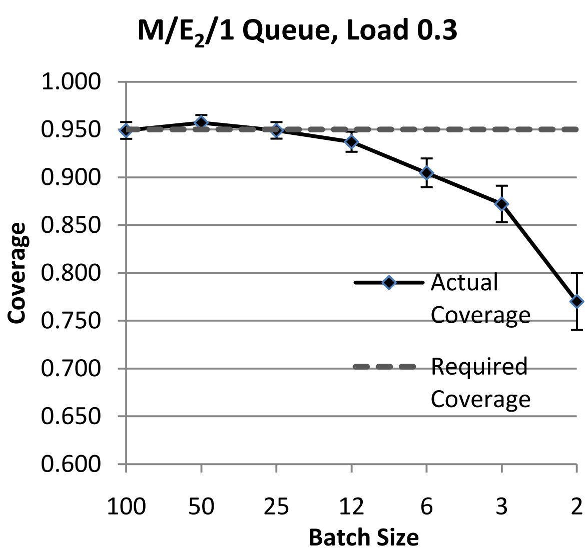

Figure 20 - Coverage of $\hat{\sigma}^2_{BM}$ with varying batch sizes on M/E$_2$/1 queue with Load 0.3

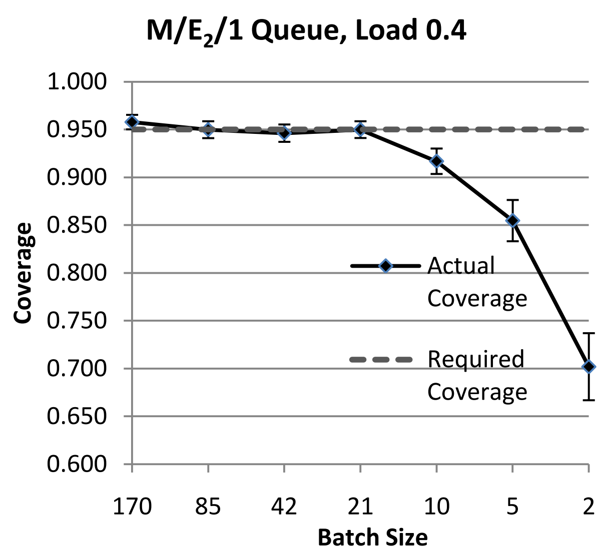

Figure 21 - Coverage of $\hat{\sigma}^2_{BM}$ with varying batch sizes on M/E$_2$/1 queue with Load 0.4

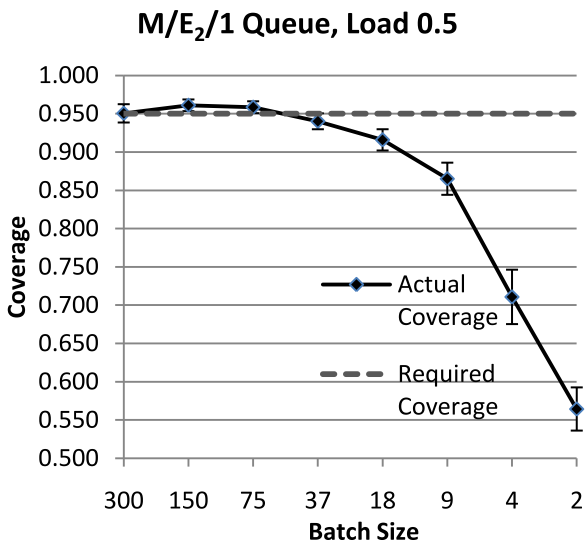

Figure 22 - Coverage of $\hat{\sigma}^2_{BM}$ with varying batch sizes on M/E$_2$/1 queue with Load 0.5

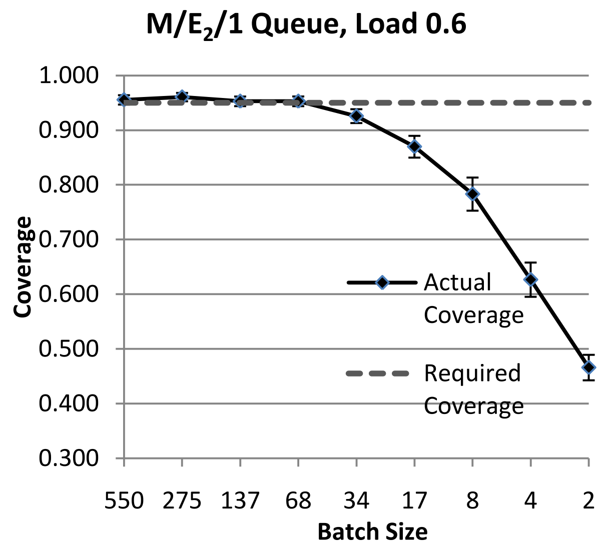

Figure 23 - Coverage of $\hat{\sigma}^2_{BM}$ with varying batch sizes on M/E$_2$/1 queue with Load 0.6

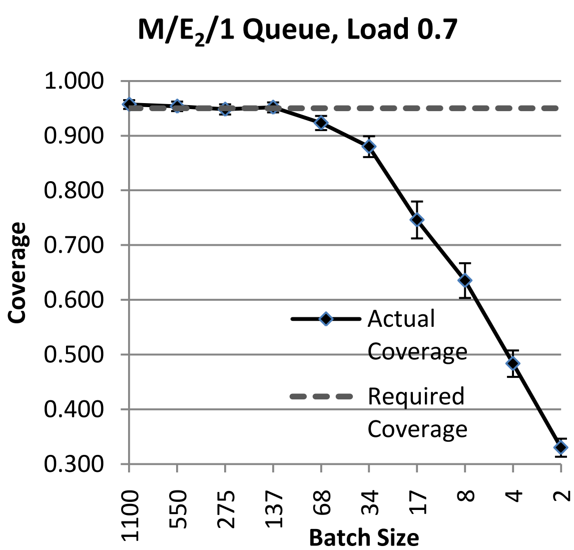

Figure 24 - Coverage of $\hat{\sigma}^2_{BM}$ with varying batch sizes on M/E$_2$/1 queue with Load 0.7

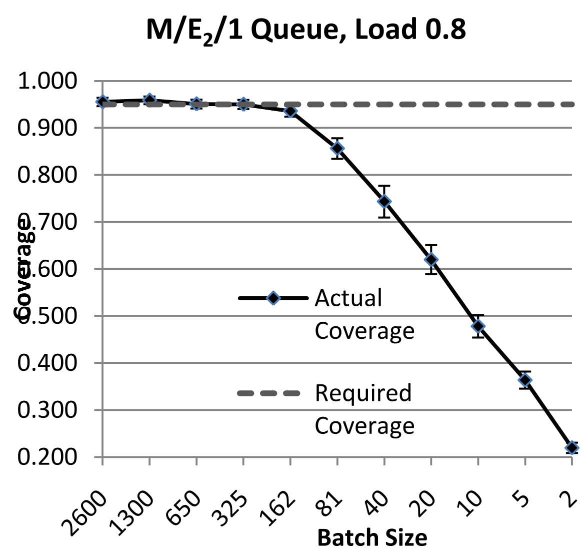

Figure 25 - Coverage of $\hat{\sigma}^2_{BM}$ with varying batch sizes on M/E$_2$/1 queue with Load 0.8

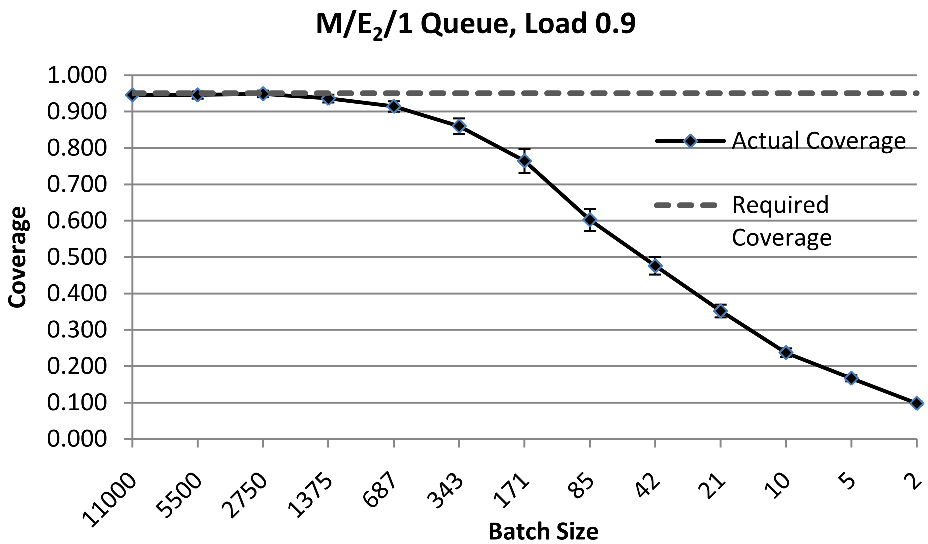

Figure 26 - Coverage of $\hat{\sigma}^2_{BM}$ with varying batch sizes on M/E_2/1 queue with Load 0.9

Once again, from these results we see that for most loads, the coverage is still adequate even after the batch size has been halved three to four times. The resulting "optimal" batch sizes for an M/E_2/1 queue are shown in Table 7.

Load	Batch Size
0.1	7
0.2	7
0.3	12
0.4	21
0.5	37
0.6	68
0.7	137
0.8	162
0.9	1375

Table 7 - Approximate Optimal Batch Sizes for Variance of Waiting Time for M/E_2/1 queue

5.3.2. M/H_2/1 Queue

The coverage of the Batch Means ($\hat{\sigma}^2_{BM}$) estimator on the M/H_2/1 queue with varying loads and batch sizes is shown in Figure 18 through to Figure 26. In all of these cases, an initial transient period of 20,000 observations is used In each of the figures, coverage over the queue with two different parameter sets is shown. The terms "Coverage C^2_x=5" and "Coverage C^2_x=50" refer to the queues with a coefficients of variation of 5 and 50 respectively.

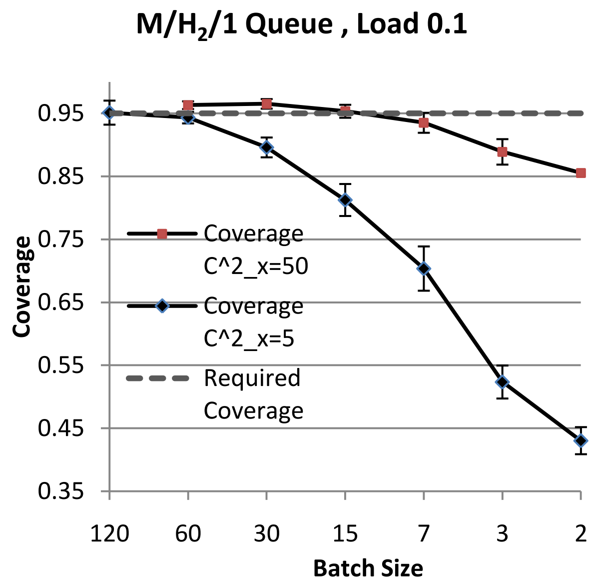

Figure 27 - Coverage of $\hat{\sigma}^2_{BM}$ with varying batch sizes two M/H_2/1 queues with differing coefficients of variation, Load 0.1

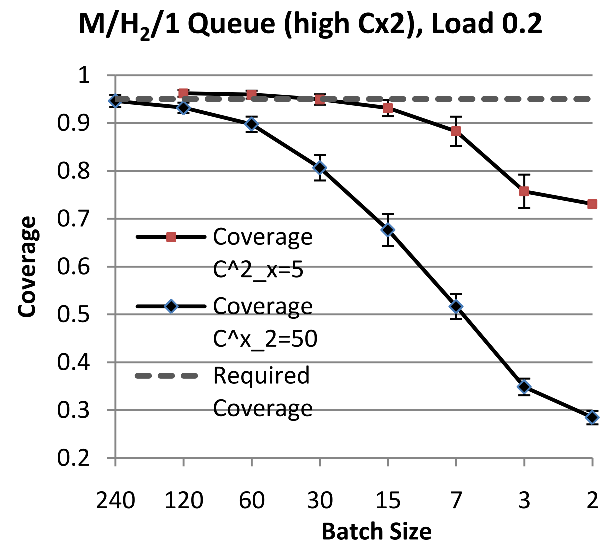

Figure 28 - Coverage of $\hat{\sigma}^2_{BM}$ with varying batch sizes two M/H_2/1 queues with differing coefficients of variation, Load 0.2

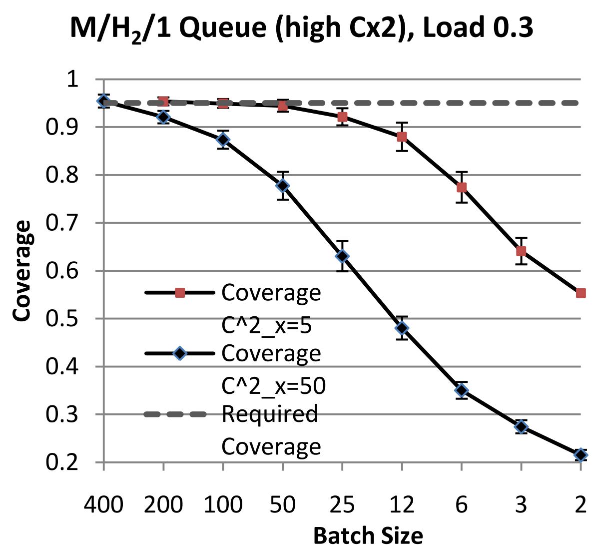

Figure 29 - Coverage of $\hat{\sigma}^2_{BM}$ with varying batch sizes two M/H_2/1 queues with differing coefficients of variation, Load 0.3

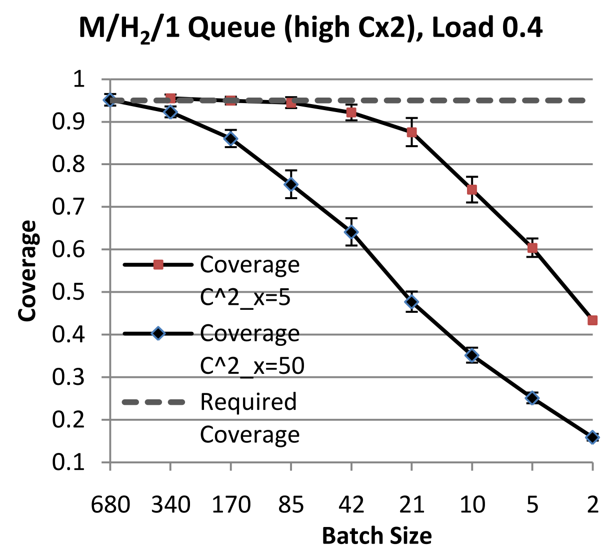

Figure 30 - Coverage of $\hat{\sigma}^2_{BM}$ with varying batch sizes two M/H_2/1 queues with differing coefficients of variation, Load 0.4

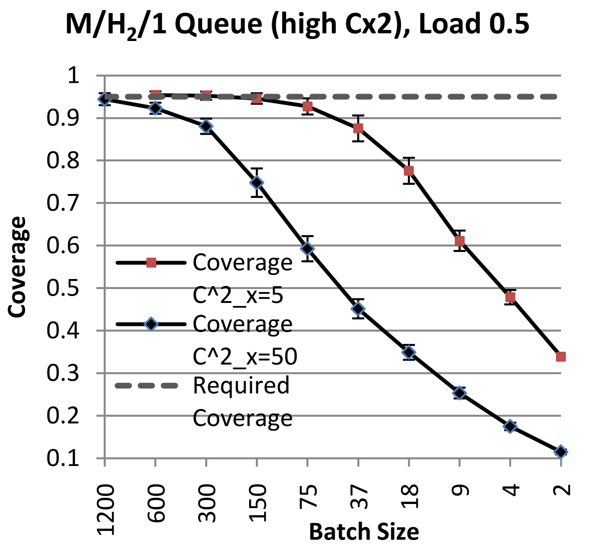

Figure 31 - Coverage of $\hat{\sigma}^2_{BM}$ with varying batch sizes two M/H_2/1 queues with differing coefficients of variation, Load 0.5

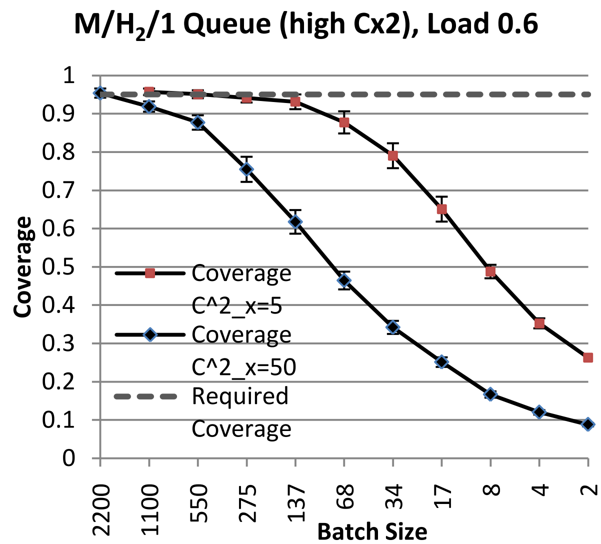

Figure 32 - Coverage of $\hat{\sigma}^2_{BM}$ with varying batch sizes two M/H_2/1 queues with differing coefficients of variation, Load 0.6

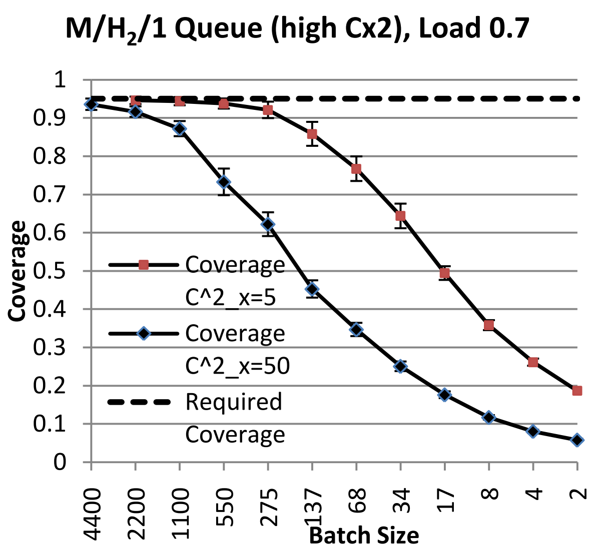

Figure 33 - Coverage of $\hat{\sigma}^2_{BM}$ with varying batch sizes two M/H_2/1 queues with differing coefficients of variation, Load 0.7

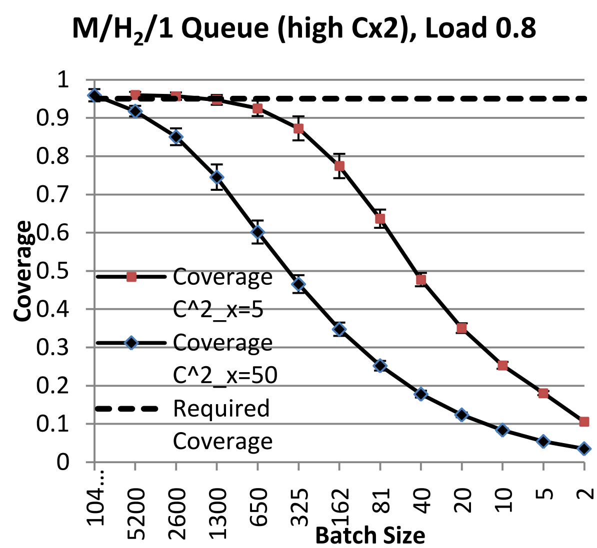

Figure 34 - Coverage of $\hat{\sigma}^2_{BM}$ with varying batch sizes two M/H_2/1 queues with differing coefficients of variation, Load 0.8

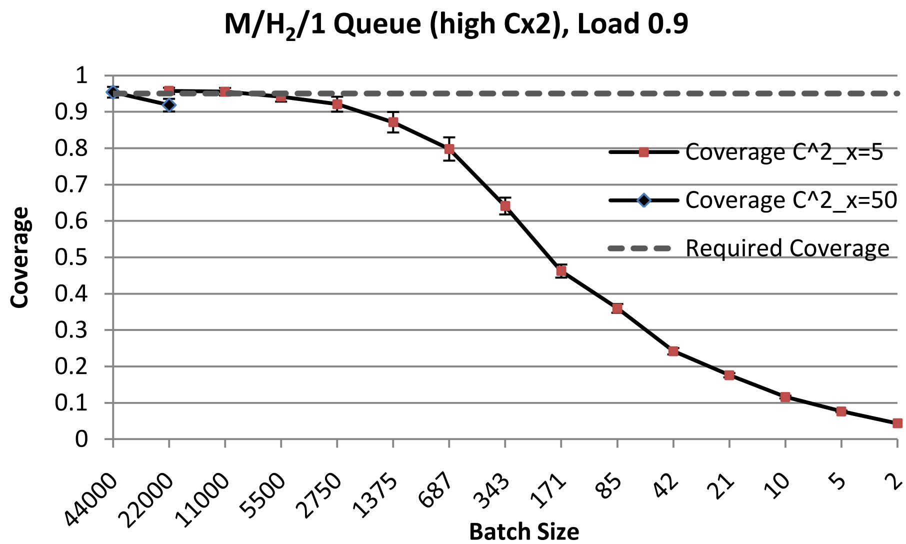

Figure 35 - Coverage of $\hat{\sigma}^2_{BM}$ with varying batch sizes two M/H_2/1 queues with differing coefficients of variation, Load 0.9

In each Figure shown, there is an instance of the Batch Means estimator providing satisfactory coverage for both versions of the queue (when a sufficient batch size is used). Also, it is interesting to note that the coverage drops off much more quickly for the queue with a higher coefficient of variation. This is because the correlation between batch means is higher, which causes the generated confidence intervals to be worse. The resulting "optimal" batch sizes for an M/H2/1 queue are shown in Table 8. Note that full testing of batch sizes for the M/H2/1 queue with $c_x^2 = 50$ and a load of 0.9, in Figure 35, was not completed due to the time required (several weeks for coverage analysis of a single batch size).

Load	Batch Size, $\boldsymbol{c_x^2 = 5}$	Batch Size, $\boldsymbol{c_x^2 = 50}$
0.1	15	60
0.2	30	120
0.3	50	400
0.4	85	680
0.5	150	1200
0.6	275	2200
0.7	550	4400
0.8	1300	10400
0.9	5500	44000

Table 8 - Approximate Optimal Batch Sizes for Variance of Waiting Time for M/H_2/1 queue

6. Estimator Implementation into Akaroa2

The determination of the best terminating simulation variance estimator was easy, as there was only one candidate. Thus the Independent Replications estimator $\hat{\sigma}^2_{Term}$ was chosen to be implemented into Akaroa2 for terminating simulation.

The determination of the best steady-state estimator was a more difficult choice. Although the Independent Replications estimator $\hat{\sigma}_{IR}^2$ produced good coverage, it was eliminated due to its poor efficiency and run length. This left two estimators, one based on Variance as a Mean Value ($\hat{\sigma}_M^2$) and the other on Batch Means ($\hat{\sigma}_{BM}^2$). The $\hat{\sigma}_{BM}^2$ estimator was found to produce good coverage on every queue and load tested, if a good batch size could be found. This is the main problem with a Batch Means style estimator, as determination of a batch size is crucially important to the result given. The $\hat{\sigma}_M^2$ estimator also produced good coverage on each queue investigated, even on queues with high loads. One advantage that this estimator has is that it's easy to implement as it makes use of existing mean value analysis modules. The main point of difference between the two is that the $\hat{\sigma}_M^2$ estimator does not rely on the correct determination of an arbitrary parameter in order to work.

Once implemented into Akaroa2, the candidate must be able to work on more complex systems than the queuing systems investigated, and it is difficult to see how a good batch size could be determined in these cases for the $\hat{\sigma}_{BM}^2$ estimator. As a result of this fact, and the ease of implementation of $\hat{\sigma}_M^2$, the $\hat{\sigma}_M^2$ estimator was chosen to be implemented into Akaroa2 for steady-state simulation

6.1. Implementation Detail

Akaroa2 provides the ability for one to implement their own *variance estimation* method, to estimate the mean value and variance for creation of mean value confidence intervals [17]. This is achieved by writing a subclass of the *VarianceEstimator* class, which defines a number of functions that Akaroa2 is aware of and uses, for example *ProcessObservation(real value)* and *GetCheckpoint(Checkpoint &cp)*.

In order to implement the variance analysis, the concrete subclass of the VarianceEstimator class was configured to do both the mean value calculation and calculation of variance for creation of mean value confidence intervals, as well as the calculation of the variance and the variance of the variance. The *Checkpoint* data structure was modified, so that it has additional parameters to store the estimated variance and variance of the variance. Lastly, the global analyser was modified to incorporate analysis of variance. The estimation of variance is treated as a mean value by the global analyser, in the same way that estimation of the mean is treated. The variance is then analysed as

$$\hat{\sigma}^2 = \frac{\sum_i n_i \hat{\sigma}_i^2}{\sum_i n_i},$$

where $\hat{\sigma}_i^2$is the local point estimate of the variance, and n_i the local number of observations, from engine i. The variance of the global estimate (variance of the variance) is calculated as

$$Var[\hat{\sigma}^2] = \frac{\sum_i n_i^2 Var[\hat{\sigma}_i^2]}{\left(\sum_i n_i^2\right)^2},$$

where $Var[\hat{\sigma}_i^2]$ is the variance of the local variance estimate at engine i.

For the terminating simulation variance estimator, the concrete subclass related to the independent replications terminating simulation module was modified. For the steady-state variance estimator, the existing concrete subclass which estimates the mean and variance of the mean by way of spectral analysis was modified.

Screenshots of the graphical user interface of Akaroa2 incorporating variance analysis are shown in Figure 36, Figure 37 and Figure 38. Figure 36 shows the "New Simulation" window, where the parameters for the simulation are set. In this case, the simulation is running until the mean-value and variance both converge to 5% error at the 95% confidence level. Figure 37 shows the simulation in progress. The red and blue lines represent the mean and variance's respective convergence to their relative errors. The Global Estimates table gives information regarding the current global estimate of parameters, for example, mean-value and its relative error, variance and its relative error. Figure 38 shows the final output of the simulation, giving information on the mean (Mean) and its confidence interval half-width (Delta), as well as the variance (Variance) and its confidence interval half-width (Delta Variance).

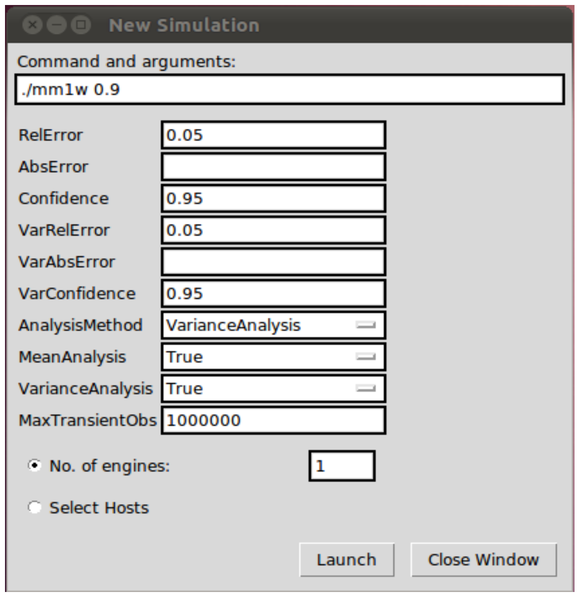

Figure 36 – Akaroa2 New Simulation Window

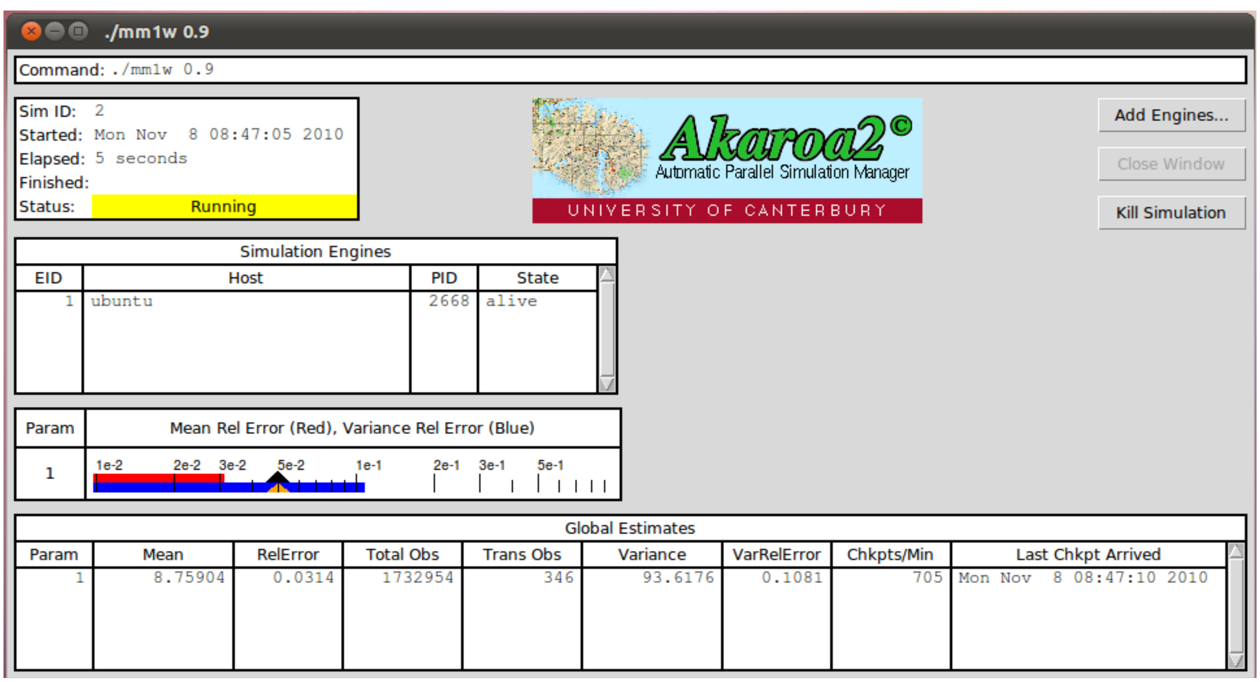

Figure 37 – Akaroa2 Simulation Running Window

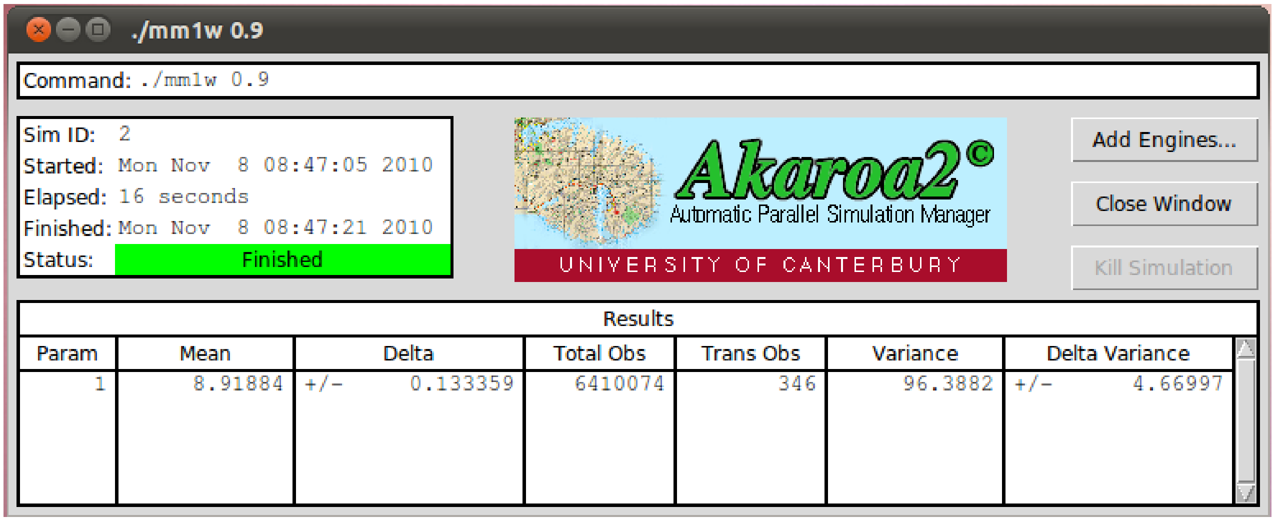

Figure 38 – Akaroa2 Finished Simulation Window

7. Conclusion

The methodology of Independent Replications works well for estimating variance in terminating simulation. It has been implemented into Akaroa2 as part of the Independent Replications module.

The difficulty of estimating variance lies in deriving accurate methodologies for steady-state simulation. Tests were run on three different estimators, which quickly identified the method of Independent Replications as being too inefficient. The remaining two estimators, based on Variance as a Mean Value and Batch Means respectively, were thoroughly tested. Variance as a Mean Value produced satisfactory coverage and was noted as being relatively easy to implement. The Batch Means estimator was found to produce good coverage if a suitable batch size was used. Efficient batch sizes were derived for different queues and loads. The inherent problem of Batch Means lies in determining the batch size, which will inevitably become more complicated as the complexity of the simulated model increases. As the Variance as a Mean Value estimator produces good coverage and does not require determination of any initial parameters, it was the estimator implemented into Akaroa2.

8. Future Work

Future work lies in the area of testing the Variance as a Mean Value and Batch Means estimators on more complicated models, such as a queuing network. Although the Variance as a Mean Value estimator was found to be the most suitable estimator to be implemented in these circumstances, if a robust batch size determination algorithm were found then the Batch Means algorithm may also be suitable.

9. References

[1] Ewing, G., K. Pawlikowski and D. McNickle. "Akaroa 2: Exploiting Network Computing by Distributed Stochastic Simulation". Proceedings of the European Simulation Multiconference, ESM'99 (Warsaw, Poland, June 1999). ISCS, 1999, 175-181.

[2] Schmidt, A. and Pawlikowski, K. and McNickle, D. Sequential Estimation of Variance in Steady-State Simulation TR-COSC 03/08. Department of Computer Science and Software Engineering, University of Canterbury, New Zealand, 2008.

[3] Law, Averill M. and Kelton, W. David. *Simulation, Modelling and Analysis*. McGraw-Hill, 2nd edition, 1991.

[4] Pawlikowski, Krzysztof. Steady-state simulation of queuing processes: A survey of problems and solutions. ACM *Computing Surveys*, 22(2): 123–170, 1990.

[5] Wilks, Samuel S. *Mathematical Statistics*. John Wiley & Sons, 2nd edition, 1962.

[6] Trivedi, Kishor S*. Probability and Statistics with Reliability, Queuing, and Computer Science Applications*. John Wiley & Sons, 2nd edition, 2002.

[7] K. Pawlikowski, D. McNickle, G. Ewing, Coverage of confidence intervals in sequential steady-state simulation, J. Simul. Pract. Theory 6 (3) (1998) 255–267 (Plus: "Erratum" 7(1):1, 1999).

[8] D. Goldsman and K.Kang, Cramer-von Mises Variance Estimators for Simulations, in Proc. 1991 Winter Simulation Conf. (IEEE Press, 1991) 916-920

[9] A. M. Law and W. D. Kelton, Confidence Inetervals for Steady-State Simulations, II: A Survey of Sequential Procedures, *Management Sci*. (1982) 550-562.

[10] Pawlikowski, Krzysztof, Yau, Victor, and McNickle, Don. Distributed stochastic discrete-event simulation in parallel time streams. *Proceedings of the 1994 Winter Simulation Conference*, (1994) 723–730.

[11] Pawlikowski, Krzysztof and McNickle, Don. Speeding up stochastic discrete-event simulation. In *Proceedings of the European Simulation Symposium, ESS 2001, Marseille, France,* pages 132–138. Oct 2001.

[12] Heidelberger, Philip and Welch, Peter D. A spectral method for confidence interval generation and run length control in simulations. *Communications of the ACM*, 25:233–245, 1981.

[13] Feldman, Richard M., Deuermeyer, Bryan L., and Yang, Yoonjung. Estimation of process variance by simulation. Unpublished paper, 1996.

[14] Gross, Donald and Harris, Carl M. *Fundamentals of Queuing Theory*. John Wiley & Sons, 2nd edition, 1985.

[15] Daley, D. J. The serial correlation coefficients of waiting times in a stationary single server queue. *J. Aust. Math. Soc. 8, 683-699* (1968).

[16] Law, A. M. Confidence intervals in discrete event simulation: a comparison of replication and batch means. *Naval Res. Logist. Quart., 24, 667-678* (1977).

[17] Ewing, G. and Pawlikowski, K. and McNickle, D. *Akaroa 2.5 User's Manual*. Technical Report TR-COSC 07/98. Department of Computer Science, University of Canterbury, New Zealand (1998).

[18] McNickle, D. C. *Personal Communication*. (2010)

[19] Pawlikowski, Krzysztof, Ewing, Greg, and McNickle, Don. Coverage of confidence intervals in sequential steady-state simulation. Journal of Simulation Practice and Theory, 6(3):255–267, 1998.

[20] Eickhoff, Mirko. *Sequential Analysis of Quantiles and Probability Distributions by Replicated Simulations* PHD 02/08. Department of Computer Science and Software Engineering, University of Canterbury, New Zealand, 2008.

Appendix

The results presented in this paper were created in C++, using the Akaroa2 framework [1]. Once data was obtained from the C++ program, it was sent to Microsoft Excel for the figure creation.

The tables below show the data used to create the figures in Section 5 (Results). The following heading symbols are used:

ρ	System load
N_r	Total number of replicated simulation runs
$\bar{l}_r$	Mean number of observations per run
N_a	Number of runs of acceptable length
p_d	Proportion of discarded runs of too short length
$\bar{c}$	Mean coverage
Δ_c	Half width of confidence interval (95% confidence level)
l_{it}	Length of initial transient period used ($\hat{\sigma}_{IR}^2$ only)

ρ	l_{it}	N_r	$\bar{l}_r$	N_a	p_d	$\bar{c}$	Δ_c
0.1	4	2872	89979	2436	0.152	0.941	0.00940
0.2	6	2126	44336	1799	0.154	0.929	0.01188
0.3	10	2782	29344	2366	0.150	0.940	0.00957
0.4	16	2087	22053	1770	0.152	0.924	0.01233
0.5	27	2255	17847	1910	0.153	0.931	0.01134
0.6	49	2573	15265	2192	0.148	0.941	0.00985
0.7	99	2445	13574	2073	0.152	0.937	0.01044
0.8	252	2247	12572	1926	0.143	0.930	0.01140
0.9	1130	2327	12025	1975	0.151	0.936	0.01078

Table 9 – Coverage Analysis Results, Estimator $\hat{\sigma}_{IR}^2$ on M/M/1 Queue

ρ	l_{it}	N_r	$\bar{l}_r$	N_a	p_d	$\bar{c}$	Δ_c
0.1	500	3848	70657	3275	0.149	0.951	0.00741
0.2	500	3572	36440	3007	0.158	0.951	0.00771
0.3	500	3415	25175	2896	0.152	0.949	0.00802
0.4	500	3672	19755	3133	0.147	0.952	0.00748
0.5	500	3779	16642	3207	0.151	0.955	0.00719
0.6	500	3791	14710	3215	0.152	0.958	0.00696
0.7	500	3470	13467	2947	0.151	0.955	0.00750
0.8	1000	4118	12685	3507	0.148	0.965	0.00611
0.9	1000	3237	12106	2744	0.152	0.955	0.00774

Table 10 – Coverage Analysis Results, Estimator $\hat{\sigma}^2_{IR}$ on M/E$_2$/1 Queue

ρ	l_{it}	N_r	$\bar{l}_r$	N_a	p_d	$\bar{c}$	Δ_c
0.1	500	3528	168953	2989	0.153	0.952	0.00768
0.2	500	3942	79179	3353	0.149	0.956	0.00695
0.3	500	4131	49336	3549	0.141	0.957	0.00668
0.4	500	3781	34610	3214	0.150	0.954	0.00725
0.5	1000	4000	26004	3388	0.153	0.960	0.00659
0.6	1000	4503	20403	3822	0.151	0.964	0.00592
0.7	1000	3721	16672	3180	0.145	0.953	0.00737
0.8	1500	3540	14181	3024	0.146	0.958	0.00712
0.9	5000	3707	12656	3144	0.152	0.955	0.00724

Table 11 - Coverage Analysis Results, Estimator $\hat{\sigma}^2_{IR}$ on M/M/1 Queue

ρ	N_r	$\bar{l}_r$	N_a	p_d	$\bar{c}$	Δ_c
0.1	3330	177872	3077	0.076	0.949	0.00778
0.2	3257	147403	2978	0.086	0.951	0.00778
0.3	3698	158336	3392	0.083	0.955	0.00701
0.4	3083	193682	2802	0.091	0.951	0.00796
0.5	3161	265180	2933	0.072	0.951	0.00780
0.6	3331	407569	3054	0.083	0.952	0.00757
0.7	3062	729058	2816	0.080	0.952	0.00792
0.8	3005	1684400	2781	0.075	0.953	0.00785
0.9	2742	7122630	2525	0.079	0.951	0.00843
0.95	2726	29591800	2533	0.071	0.953	0.00821

Table 12 - Coverage Analysis Results, Estimator $\hat{\sigma}^2_{M}$ on M/M/1 Queue

ρ	N_r	$\bar{l}_r$	N_a	p_d	$\bar{c}$	Δ_c
0.1	4138	129988	3850	0.070	0.955	0.00653
0.2	3412	105831	3133	0.082	0.952	0.00748
0.3	3268	113680	2994	0.084	0.952	0.00767
0.4	3456	141514	3170	0.083	0.957	0.00703
0.5	3173	191346	2878	0.093	0.957	0.00739
0.6	3160	301227	2873	0.091	0.956	0.00752
0.7	3429	547812	3098	0.097	0.957	0.00716
0.8	2708	1267700	2469	0.088	0.950	0.00862
0.9	2729	5415920	2527	0.074	0.947	0.00871
0.95	3473	22797400	3180	0.084	0.960	0.00776

Table 13 - Coverage Analysis Results, Estimator $\hat{\sigma}^2_{M}$ on M/E$_2$/1 Queue

ρ	N_r	$\bar{l}_r$	N_a	p_d	$\bar{c}$	Δ_c
0.1	3965	662148	3608	0.090	0.956	0.00668
0.2	4229	636621	3841	0.092	0.959	0.00630
0.3	3197	709281	2907	0.091	0.947	0.00812
0.4	3226	849411	2940	0.089	0.952	0.00773
0.5	2975	1095830	2716	0.087	0.946	0.00851
0.6	2873	1564390	2639	0.081	0.946	0.00864
0.7	2776	2568080	2545	0.083	0.948	0.00862
0.8	2851	5556790	2608	0.085	0.950	0.00838
0.9	2332	21705200	2152	0.077	0.949	0.00927
0.95	4326	108044000	3986	0.079	0.958	0.00814

Table 14 - Coverage Analysis Results, Estimator $\hat{\sigma}_M^2$ on M/H_2/1 (Cx2 = 5) Queue

ρ	N_r	$\bar{l}_r$	N_a	p_d	$\bar{c}$	Δ_c
0.1	3845	4394970	3499	0.090	0.957	0.00671
0.2	3826	4969750	3537	0.076	0.957	0.00671
0.3	3367	5878780	3103	0.078	0.952	0.00755
0.4	4058	7619400	3732	0.080	0.958	0.00642
0.5	3803	11102300	3313	0.129	0.962	0.00654
0.6	3566	16056400	3113	0.127	0.949	0.00630
0.7	2613	25119200	2393	0.084	0.933	0.00758
0.8	4300	52140400	3993	0.071	0.965	0.00615
0.9	3281	92105350	2977	0.093	0.948	0.00775
0.95	2985	735000000	2757	0.076	0.940	0.00650

Table 15 - Coverage Analysis Results, Estimator $\hat{\sigma}_M^2$ on M/H_2/1 (Cx2 = 50) Queue

ρ	Batch Size	N_r	$\bar{l}_r$	N_a	p_d	$\bar{c}$	Δ_c
0.1	30	3215	155266	2864	0.109	0.954	0.00765
0.1	15	3153	152186	2782	0.118	0.950	0.00810
0.1	7	2658	145401	2303	0.134	0.937	0.00995
0.1	3	2328	127808	2016	0.134	0.929	0.01121
0.1	2	789	46787	765	0.030	0.769	0.02989

Table 16 - Coverage Analysis Results, Estimator $\hat{\sigma}_{BM}^2$ on M/M/1 Queue

ρ	Batch Size	N_r	$\bar{l}_r$	N_a	p_d	$\bar{c}$	Δ_c
0.2	60	3134	125999	2835	0.095	0.957	0.00750
0.2	30	2695	123351	2419	0.102	0.947	0.00889
0.2	15	2736	117678	2443	0.107	0.950	0.00867
0.2	7	2164	105251	1946	0.101	0.930	0.01133
0.2	3	1463	79274	1347	0.079	0.884	0.01709
0.2	2	1030	63667	995	0.034	0.832	0.02323

Table 17 - Coverage Analysis Results, Estimator $\hat{\sigma}_{BM}^2$ on M/M/1 Queue

ρ	Batch Size	N_r	$\bar{l}_r$	N_a	p_d	$\bar{c}$	Δ_c
0.3	100	2994	134599	2747	0.082	0.957	0.00761
0.3	50	2922	131467	2632	0.099	0.952	0.00819
0.3	25	2555	124934	2307	0.097	0.948	0.00903
0.3	12	2114	111623	1912	0.096	0.931	0.01133
0.3	6	1600	89266	1451	0.093	0.899	0.01548
0.3	3	1034	62203	949	0.082	0.832	0.02377
0.3	2	789	46787	765	0.030	0.769	0.02989

Table 18 - Coverage Analysis Results, Estimator $\hat{\sigma}^2_{BM}$ on M/M/1 Queue

ρ	Batch Size	N_r	$\bar{l}_r$	N_a	p_d	$\bar{c}$	Δ_c
0.4	170	2902	164362	2638	0.091	0.958	0.00770
0.4	85	2824	160929	2567	0.091	0.955	0.00804
0.4	42	2633	153716	2355	0.106	0.947	0.00902
0.4	21	2405	138453	2119	0.119	0.938	0.01029
0.4	10	1416	108769	1275	0.100	0.884	0.01759
0.4	5	1018	75256	890	0.126	0.831	0.02460
0.4	2	903	36657	858	0.050	0.642	0.03208

Table 19 - Coverage Analysis Results, Estimator $\hat{\sigma}^2_{BM}$ on M/M/1 Queue

ρ	Batch Size	N_r	$\bar{l}_r$	N_a	p_d	$\bar{c}$	Δ_c
0.5	300	2550	224081	2336	0.084	0.955	0.00840
0.5	150	2685	219539	2431	0.095	0.955	0.00826
0.5	75	2126	210916	1912	0.101	0.940	0.01062
0.5	37	2137	190981	1901	0.110	0.938	0.01085
0.5	18	1619	152865	1444	0.108	0.910	0.01477
0.5	9	879	105353	818	0.069	0.811	0.02686
0.5	4	665	58238	616	0.074	0.714	0.03568
0.5	2	1304	30341	1225	0.061	0.557	0.02782

Table 20 - Coverage Analysis Results, Estimator $\hat{\sigma}^2_{BM}$ on M/M/1 Queue

ρ	Batch Size	N_r	$\bar{l}_r$	N_a	p_d	$\bar{c}$	Δ_c
0.6	550	2521	347626	2303	0.086	0.953	0.00864
0.6	275	3063	340703	2803	0.085	0.960	0.00722
0.6	137	2558	325159	2337	0.086	0.952	0.00866
0.6	68	2324	294472	2100	0.096	0.939	0.01027
0.6	34	1487	238610	1346	0.095	0.897	0.01626
0.6	17	1025	166587	937	0.086	0.840	0.02348
0.6	8	843	96270	797	0.055	0.659	0.03292
0.6	4	1332	53042	1241	0.068	0.554	0.02766
0.6	2	2437	26258	2277	0.066	0.403	0.02015

Table 21 - Coverage Analysis Results, Estimator $\hat{\sigma}^2_{BM}$ on M/M/1 Queue

\

ρ	Batch Size	N_r	$\bar{l}_r$	N_a	p_d	$\bar{c}$	Δ_c
0.7	1100	2809	617967	2555	0.090	0.959	0.00766
0.7	550	2292	600315	2078	0.093	0.950	0.00938
0.7	275	2450	574562	2229	0.090	0.955	0.00864
0.7	137	1967	520737	1744	0.113	0.938	0.01131
0.7	68	1529	420172	1378	0.099	0.909	0.01522
0.7	34	1011	291326	934	0.076	0.837	0.02368
0.7	17	722	178365	668	0.075	0.760	0.03237
0.7	8	1185	92475	1078	0.090	0.588	0.02939

Table 22 - Coverage Analysis Results, Estimator $\hat{\sigma}^2_{BM}$ on M/M/1 Queue

ρ	Batch Size	N_r	$\bar{l}_r$	N_a	p_d	$\bar{c}$	Δ_c
0.8	2600	2314	1426050	2136	0.077	0.952	0.00909
0.8	1300	2422	1399540	2200	0.092	0.955	0.00862
0.8	650	2402	1333680	2163	0.100	0.956	0.00868
0.8	325	2164	1196100	1955	0.097	0.938	0.01072
0.8	162	1610	964145	1443	0.104	0.910	0.01478
0.8	81	975	663714	898	0.079	0.825	0.02485
0.8	40	822	391118	749	0.089	0.673	0.03361
0.8	20	1472	212514	1401	0.048	0.523	0.02616
0.8	10	2242	109835	2129	0.050	0.419	0.02097

Table 23 - Coverage Analysis Results, Estimator $\hat{\sigma}^2_{BM}$ on M/M/1 Queue

ρ	Batch Size	N_r	$\bar{l}_r$	N_a	p_d	$\bar{c}$	Δ_c
0.9	11000	2307	6094930	2086	0.096	0.954	0.00904
0.9	5500	2479	5950270	2242	0.096	0.958	0.00834
0.9	2750	2329	5637300	2094	0.101	0.951	0.00926
0.9	1375	1907	5013610	1698	0.110	0.938	0.01151
0.9	687	1455	4003360	1296	0.109	0.905	0.01596
0.9	343	926	2694900	855	0.077	0.814	0.02609
0.9	171	755	1609260	685	0.093	0.692	0.03458
0.9	85	1203	869323	1129	0.062	0.577	0.02883

Table 24 - Coverage Analysis Results, Estimator $\hat{\sigma}^2_{BM}$ on M/M/1 Queue

ρ	Batch Size	N_r	$\bar{l}_r$	N_a	p_d	$\bar{c}$	Δ_c
0.1	30	3303	111780	3019	0.086	0.952	0.00760
0.1	15	3451	110086	3096	0.103	0.953	0.00747
0.1	7	3615	105868	3220	0.109	0.950	0.00755
0.1	3	2577	95107	2250	0.127	0.936	0.01015
0.1	2	2205	86596	2072	0.060	0.917	0.01191

Table 25 - Coverage Analysis Results, Estimator $\hat{\sigma}^2_{BM}$ on M/E_2/1 Queue

ρ	Batch Size	N_r	$\bar{l}_r$	N_a	p_d	$\bar{c}$	Δ_c
0.2	60	3275	89789	3016	0.079	0.958	0.00720
0.2	30	2644	88190	2644	0.000	0.951	0.00821
0.2	15	3022	85101	2736	0.095	0.954	0.00789
0.2	7	2369	77447	2109	0.110	0.936	0.01048
0.2	3	1809	61532	1581	0.126	0.908	0.01423
0.2	2	1268	50871	1224	0.035	0.859	0.01947

Table 26 - Coverage Analysis Results, Estimator $\hat{\sigma}^2_{BM}$ on M/E$_2$/1 Queue

ρ	Batch Size	N_r	$\bar{l}_r$	N_a	p_d	$\bar{c}$	Δ_c
0.3	100	2658	96028	2411	0.093	0.949	0.00875
0.3	50	2848	93978	2572	0.097	0.957	0.00782
0.3	25	2764	90752	2472	0.106	0.949	0.00867
0.3	12	2313	82772	2053	0.112	0.937	0.01050
0.3	6	1684	68955	1468	0.128	0.905	0.01503
0.3	3	1282	49808	1172	0.086	0.872	0.01913
0.3	2	801	38634	774	0.034	0.770	0.02965

Table 27 - Coverage Analysis Results, Estimator $\hat{\sigma}^2_{BM}$ on M/E$_2$/1 Queue

ρ	Batch Size	N_r	$\bar{l}_r$	N_a	p_d	$\bar{c}$	Δ_c
0.4	170	3029	120035	2716	0.103	0.958	0.00757
0.4	85	2347	117372	2347	0.000	0.950	0.00884
0.4	42	2613	112910	2375	0.091	0.946	0.00908
0.4	21	2666	104209	2390	0.104	0.950	0.00876
0.4	10	1858	85739	1657	0.108	0.917	0.01331
0.4	5	1145	61692	1025	0.105	0.855	0.02158
0.4	2	687	31749	654	0.048	0.702	0.03507

Table 28 - Coverage Analysis Results, Estimator $\hat{\sigma}^2_{BM}$ on M/E$_2$/1 Queue

ρ	Batch Size	N_r	$\bar{l}_r$	N_a	p_d	$\bar{c}$	Δ_c
0.5	300	2169	164815	1964	0.095	0.951	0.01196
0.5	150	2699	161606	2468	0.086	0.961	0.00763
0.5	75	2819	156341	2535	0.101	0.959	0.00776
0.5	37	2238	144487	1990	0.111	0.940	0.01042
0.5	18	1729	121781	1535	0.112	0.916	0.01388
0.5	9	1157	89147	1016	0.122	0.865	0.02101
0.5	4	689	51571	626	0.091	0.711	0.03552
0.5	2	1250	27838	1187	0.050	0.564	0.02821

Table 29 - Coverage Analysis Results, Estimator $\hat{\sigma}^2_{BM}$ on M/E$_2$/1 Queue

ρ	Batch Size	N_r	$\bar{l}_r$	N_a	p_d	$\bar{c}$	Δ_c
0.6	550	2455	258620	2220.0	0.096	0.955	0.00859
0.6	275	2755	254285	2480.0	0.100	0.960	0.00767
0.6	137	2491	244915	2223.0	0.108	0.953	0.00882
0.6	68	2429	226002	2182.0	0.102	0.953	0.00890
0.6	34	1852	191904	1642.0	0.113	0.926	0.01269
0.6	17	1210	140940	1098.0	0.093	0.870	0.01991
0.6	8	782	85891	710.0	0.092	0.783	0.03032
0.6	4	1016	48371	916.0	0.098	0.627	0.03133
0.6	2	1860	24705	1762.0	0.053	0.466	0.02330

Table 30 - Coverage Analysis Results, Estimator $\hat{\sigma}^2_{BM}$ on M/E_2/1 Queue

ρ	Batch Size	N_r	$\bar{l}_r$	N_a	p_d	$\bar{c}$	Δ_c
0.7	1100	2720	466971	2464	0.094	0.957	0.00801
0.7	550	2452	457365	2243	0.085	0.954	0.00870
0.7	275	2427	441961	2194	0.096	0.948	0.00929
0.7	137	2333	409594	2113	0.094	0.952	0.00914
0.7	68	1831	349922	1641	0.104	0.923	0.01288
0.7	34	1242	258200	1125	0.094	0.880	0.01899
0.7	17	706	164934	642	0.091	0.746	0.03367
0.7	8	947	88861	883	0.068	0.635	0.03175
0.7	4	1762	46399	1642	0.068	0.484	0.02418
0.7	2	3327	22381	3121	0.062	0.330	0.01650

Table 31 - Coverage Analysis Results, Estimator $\hat{\sigma}^2_{BM}$ on M/E_2/1 Queue

ρ	Batch Size	N_r	$\bar{l}_r$	N_a	p_d	$\bar{c}$	Δ_c
0.8	2600	2273	1082990	2058	0.095	0.955	0.00893
0.8	1300	2489	1067600	2255	0.094	0.959	0.00821
0.8	650	2261	1027180	2056	0.091	0.951	0.00934
0.8	325	2176	945447	1941	0.108	0.950	0.00970
0.8	162	1887	798707	1682	0.109	0.936	0.01172
0.8	81	1082	580804	988	0.087	0.856	0.02188
0.8	40	688	363553	639	0.071	0.743	0.03387
0.8	20	1015	205163	944	0.070	0.620	0.030975
0.8	10	1813	106841	1680	0.073	0.478	0.02389
0.8	5	2922	53462	2693	0.078	0.364	0.01817
0.8	2	5854	19218	5474	0.065	0.219	0.01097

Table 32 - Coverage Analysis Results, Estimator $\hat{\sigma}^2_{BM}$ on M/E_2/1 Queue

$\boldsymbol{\rho}$	Batch Size	N_r	$\bar{l}_r$	N_a	p_d	$\bar{c}$	Δ_c
0.9	11000	2265	4624970	2074	0.084	0.946	0.00098
0.9	5500	2238	4544160	2041	0.088	0.946	0.00984
0.9	2750	2440	4396750	2200	0.098	0.949	0.00923
0.9	1375	2170	4038760	1983	0.086	0.936	0.01078
0.9	687	1652	3358220	1502	0.091	0.914	0.01417
0.9	343	1142	2441310	1037	0.092	0.860	0.02111
0.9	171	724	1519760	641	0.115	0.764	0.03286
0.9	85	1098	833570	1016	0.075	0.602	0.03010
0.9	42	1823	431365	1694	0.071	0.476	0.02379
0.9	21	3129	215492	2834	0.094	0.352	0.01759
0.9	10	5204	96594	4946	0.050	0.237	0.01186
0.9	5	8235	43607	7694	0.066	0.167	0.00833
0.9	2	15120	13538	14187	0.062	0.098	0.00489

Table 33 - Coverage Analysis Results, Estimator $\hat{\sigma}^2_{BM}$ on M/E_2/1 Queue

$\boldsymbol{\rho}$	Batch Size	N_r	$\bar{l}_r$	N_a	p_d	$\bar{c}$	Δ_c
0.1	60	4059	566184	3604	0.112	0.963	0.00616
0.1	30	3874	550797	3445	0.111	0.965	0.00612
0.1	15	3313	521258	2939	0.113	0.953	0.00762
0.1	7	2462	453746	2189	0.111	0.935	0.01032
0.1	3	1615	324770	1504	0.069	0.889	0.01588
0.1	2	1195	254482	1162	0.028	0.855	0.02022

Table 34 - Coverage Analysis Results, Estimator $\hat{\sigma}^2_{BM}$ on M/H_2/1 (Cx2=5) Queue

$\boldsymbol{\rho}$	Batch Size	N_r	$\bar{l}_r$	N_a	p_d	$\bar{c}$	Δ_c
0.2	120	3671	539554	3299	0.101	0.96211	0.00652
0.2	60	3483	524160	3113	0.106	0.960	0.00692
0.2	30	3030	493726	2685	0.114	0.949	0.00830
0.2	15	2345	430726	2086	0.110	0.931	0.01085
0.2	7	1483	321054	1350	0.090	0.883	0.01715
0.2	3	796	185855	758	0.048	0.757	0.03053
0.2	2	649	130730	613	0.055	0.731	0.03512

Table 35- Coverage Analysis Results, Estimator $\hat{\sigma}^2_{BM}$ on M/H_2/1 (Cx2=5) Queue

ρ	Batch Size	N_r	$\bar{l}_r$	N_a	p_d	$\bar{c}$	Δ_c
0.3	200	3012	599621	2708	0.101	0.953	0.00793
0.3	100	2959	580622	2671	0.097	0.949	0.00834
0.3	50	2644	540835	2370	0.104	0.944	0.00924
0.3	25	2055	467438	1815	0.117	0.921	0.01240
0.3	12	1414	345125	1278	0.096	0.879	0.01785
0.3	6	816	220348	766	0.061	0.774	0.02962
0.3	3	888	125209	863	0.028	0.641	0.03202
0.3	2	1326	83534	1242	0.063	0.553	0.02766

Table 36 - Coverage Analysis Results, Estimator $\hat{\sigma}^2_{BM}$ on M/H_2/1 (Cx2=5) Queue

ρ	Batch Size	N_r	$\bar{l}_r$	N_a	p_d	$\bar{c}$	Δ_c
0.4	340	3101	721644	2811	0.094	0.955	0.00765
0.4	170	2785	697557	2534	0.090	0.949	0.00853
0.4	85	2684	651131	2424	0.097	0.945	0.00910
0.4	42	1909	558750	1696	0.112	0.922	0.01280
0.4	21	1380	416367	1221	0.115	0.876	0.01852
0.4	10	713	255016	674	0.055	0.740	0.03311
0.4	5	1042	143503	1010	0.031	0.604	0.03017
0.4	2	2127	59539	2008	0.056	0.434	0.02168

Table 37 - Coverage Analysis Results, Estimator $\hat{\sigma}^2_{BM}$ on M/H_2/1 (Cx2=5) Queue

ρ	Batch Size	N_r	$\bar{l}_r$	N_a	p_d	$\bar{c}$	Δ_c
0.5	600	2620	927262	2374	0.094	0.954	0.00842
0.5	300	2653	899519	2341	0.118	0.953	0.00861
0.5	150	2352	840571	2089	0.112	0.946	0.00970
0.5	75	1911	725940	1692	0.115	0.927	0.01237
0.5	37	1328	542127	1158	0.128	0.876	0.01901
0.5	18	791	340960	723	0.086	0.776	0.03040
0.5	9	1047	191193	978	0.066	0.611	0.03055
0.5	4	1831	90051	1675	0.085	0.479	0.02393
0.5	2	3216	43423	2999	0.067	0.339	0.01695

Table 38 - Coverage Analysis Results, Estimator $\hat{\sigma}^2_{BM}$ on M/H_2/1 (Cx2=5) Queue

ρ	Batch Size	N_r	$\bar{l}_r$	N_a	p_d	$\bar{c}$	Δ_c
0.6	1100	2828	1336430	2555.0	0.097	0.957	0.00784
0.6	550	2641	1289990	2411.0	0.087	0.951	0.00861
0.6	275	2145	1200300	1941.0	0.095	0.941	0.01046
0.6	137	1999	1037910	1762.0	0.119	0.931	0.01181
0.6	68	1260	769393	1111.0	0.118	0.878	0.01928
0.6	34	833	493010	759.0	0.089	0.791	0.02896
0.6	17	913	281354	825.0	0.096	0.651	0.03253
0.6	8	1774	140621	1615.0	0.090	0.488	0.03253
0.6	4	3026	69886	2824.0	0.067	0.353	0.01763
0.6	2	4659	32530	4317.0	0.073	0.263	0.01313

Table 39 - Coverage Analysis Results, Estimator $\hat{\sigma}^2_{BM}$ on M/H_2/1 (Cx2=5) Queue

ρ	Batch Size	N_r	$\bar{l}_r$	N_a	p_d	$\bar{c}$	Δ_c
0.7	2200	2441	2190740	2218	0.091	0.946	0.00942
0.7	1100	2380	2111640	2169	0.089	0.944	0.00970
0.7	550	2144	1969500	1937	0.097	0.938	0.01078
0.7	275	1807	1700870	1593	0.118	0.921	0.01326
0.7	137	1111	1260880	1009	0.092	0.858	0.02152
0.7	68	748	799360	700	0.064	0.767	0.03132
0.7	34	950	457098	853	0.102	0.644	0.03215
0.7	17	1643	457098	1574	0.042	0.494	0.03215
0.7	8	2970	112831	2756	0.072	0.358	0.01790
0.7	4	4624	54070	4347	0.060	0.261	0.01306
0.7	2	7401	23867	6707	0.094	0.187	0.00932

Table 40 - Coverage Analysis Results, Estimator $\hat{\sigma}^2_{BM}$ on M/H_2/1 (Cx2=5) Queue

ρ	Batch Size	N_r	$\bar{l}_r$	N_a	p_d	$\bar{c}$	Δ_c
0.8	5200	2745	4708780	2485	0.095	0.960	0.00773
0.8	2600	2604	4547900	2334	0.104	0.957	0.00826
0.8	1300	2257	4204000	2020	0.105	0.947	0.00977
0.8	650	1825	3571720	1625	0.110	0.925	0.01282
0.8	325	1181	2618820	1044	0.116	0.873	0.02023
0.8	162	759	1649740	682	0.101	0.774	0.03139
0.8	81	949	922513	880	0.073	0.636	0.03179
0.8	40	1841	477978	1687	0.084	0.477	0.02384
0.8	20	3144	237883	2852	0.093	0.350	0.01751
0.8	10	4837	115035	4542	0.061	0.253	0.01265
0.8	5	7694	52778	6985	0.092	0.180	0.00902
0.8	2	14961	16487	13040	0.128	0.106	0.00527

Table 41 - Coverage Analysis Results, Estimator $\hat{\sigma}^2_{BM}$ on M/H_2/1 (Cx2=5) Queue

ρ	Batch Size	N_r	$\bar{l}_r$	N_a	p_d	$\bar{c}$	Δ_c
0.9	22000	2600	1863760	2333	0.103	0.958	0.00818
0.9	11000	2371	17835400	2150	0.093	0.955	0.00873
0.9	5500	2112	16378700	1921	0.090	0.942	0.01048
0.9	2750	1708	13870200	1522	0.109	0.921	0.01354
0.9	1375	1126	9947620	1020	0.094	0.872	0.02054
0.9	687	847	6217060	782	0.077	0.798	0.02815
0.9	343	916	3500090	861	0.060	0.641	0.03205
0.9	171	1907	1803060	1789	0.062	0.462	0.02311
0.9	85	2940	898957	2738	0.069	0.360	0.01798
0.9	42	5156	419345	4818	0.066	0.242	0.01210
0.9	21	7820	191677	7202	0.079	0.176	0.00880
0.9	10	13177	77201	11719	0.111	0.116	0.00580
0.9	5	21306	29993	18524	0.131	0.077	0.00383
0.9	2	40868	7294	33731	0.175	0.044	0.00218

Table 42 - Coverage Analysis Results, Estimator $\hat{\sigma}^2_{BM}$ on M/H_2/1 (Cx2=5) Queue

ρ	Batch Size	N_r	$\bar{l}_r$	N_a	p_d	$\bar{c}$	Δ_c
0.1	120	3184	3395550	2817	0.115	0.951	0.01910
0.1	60	2710	3070010	2402	0.114	0.943	0.00924
0.1	30	1633	2510540	1443	0.116	0.896	0.01575
0.1	15	1000	1747150	907	0.093	0.813	0.02540
0.1	7	673	983062	648	0.037	0.704	0.03517
0.1	3	1456	453111	1400	0.038	0.524	0.02617
0.1	2	2179	298441	2035	0.066	0.430	0.02152

Table 43 - Coverage Analysis Results, Estimator $\hat{\sigma}^2_{BM}$ on M/H_2/1 (Cx2=50) Queue

ρ	Batch Size	N_r	$\bar{l}_r$	N_a	p_d	$\bar{c}$	Δ_c
0.2	240	2718	3732220	2401	0.117	0.946	0.012402
0.2	120	2311	3324770	2040	0.117	0.932	0.010937
0.2	60	1627	2607540	1408	0.135	0.898	0.015831
0.2	30	939	1720810	863	0.081	0.806	0.026363
0.2	15	757	974017	736	0.028	0.677	0.033801
0.2	7	1498	489250	1440	0.039	0.517	0.025816
0.2	3	3023	212563	2874	0.049	0.349	0.017426
0.2	2	4175	135833	3864	0.074	0.285	0.014232

Table 44 - Coverage Analysis Results, Estimator $\hat{\sigma}^2_{BM}$ on M/H_2/1 (Cx2=50) Queue

ρ	Batch Size	N_r	$\bar{l}_r$	N_a	p_d	$\bar{c}$	Δ_c
0.3	400	2673	4428950	2382	0.109	0.954	0.01354
0.3	200	1886	3824910	1661	0.119	0.921	0.01301
0.3	100	1358	2898230	1202	0.115	0.874	0.01879
0.3	50	836	1859140	786	0.060	0.777	0.02909
0.3	25	930	1047710	903	0.029	0.630	0.03149
0.3	12	1739	527351	1666	0.042	0.480	0.02400
0.3	6	2962	268759	2853	0.037	0.350	0.01751
0.3	3	4310	131197	4072	0.055	0.274	0.01370
0.3	2	6061	82150	5609	0.075	0.215	0.01076

Table 45 - Coverage Analysis Results, Estimator $\hat{\sigma}^2_{BM}$ on M/H_2/1 (Cx2=50) Queue

ρ	Batch Size	N_r	$\bar{l}_r$	N_a	p_d	$\bar{c}$	Δ_c
0.4	680	2145	5423860	1897	0.116	0.951	0.01360
0.4	340	1907	4664070	1675	0.122	0.923	0.01277
0.4	170	1229	3475670	1132	0.079	0.860	0.02019
0.4	85	724	2231920	672	0.072	0.753	0.03262
0.4	42	909	1241530	861	0.053	0.641	0.03205
0.4	21	1768	644488	1687	0.046	0.477	0.02384
0.4	10	3009	309033	2836	0.057	0.352	0.01758
0.4	5	5141	150750	4587	0.108	0.251	0.01255
0.4	2	8963	53196	8149	0.091	0.159	0.00794

Table 46 - Coverage Analysis Results, Estimator $\hat{\sigma}^2_{BM}$ on M/H_2/1 (Cx2=50) Queue

ρ	Batch Size	N_r	$\bar{l}_r$	N_a	p_d	$\bar{c}$	Δ_c
0.5	1200	2118	7166710	1834	0.134	0.944	0.01408
0.5	600	1837	6174980	1610	0.124	0.923	0.01303
0.5	300	1403	4590250	1256	0.105	0.881	0.01794
0.5	150	708	2925680	647	0.086	0.748	0.03346
0.5	75	1143	1645050	1056	0.076	0.593	0.02964
0.5	37	1952	840952	1869	0.043	0.452	0.02257
0.5	18	2989	413646	2867	0.041	0.349	0.01745
0.5	9	4743	201797	4529	0.045	0.253	0.01267
0.5	4	8006	83346	7253	0.094	0.175	0.00875
0.5	2	12921	35895	11823	0.085	0.115	0.00575

Table 47 - Coverage Analysis Results, Estimator $\hat{\sigma}^2_{BM}$ on M/H_2/1 (Cx2=50) Queue

ρ	Batch Size	N_r	$\bar{l}_r$	N_a	p_d	$\bar{c}$	Δ_c
0.6	2200	2405	10391800	2155	0.104	0.954	0.01206
0.6	1100	1767	8816680	1590	0.100	0.919	0.01342
0.6	550	1268	6560160	1165	0.081	0.877	0.01885
0.6	275	720	4148320	661	0.082	0.755	0.03280
0.6	137	990	2333690	952	0.038	0.618	0.03088
0.6	68	1856	1208360	1774	0.044	0.464	0.02321
0.6	34	3108	607550	2959	0.048	0.342	0.01710
0.6	17	4776	293666	4580	0.041	0.251	0.01257
0.6	8	8333	128946	7692	0.077	0.167	0.00833
0.6	4	12372	57546	11261	0.090	0.120	0.00601
0.6	2	17983	23075	15927	0.114	0.088	0.00440

Table 48 - Coverage Analysis Results, Estimator $\hat{\sigma}^2_{BM}$ on M/H_2/1 (Cx2=50) Queue

ρ	Batch Size	N_r	$\bar{l}_r$	N_a	p_d	$\bar{c}$	Δ_c
0.7	4400	1858	17297200	1662	0.105	0.936	0.01490
0.7	2200	1735	14540700	1553	0.105	0.916	0.01378
0.7	1100	1228	10811100	1101	0.103	0.872	0.01974
0.7	550	664	6847360	621	0.065	0.733	0.03481
0.7	275	1033	3896950	934	0.096	0.622	0.03110
0.7	137	1993	2004510	1860	0.067	0.453	0.02263
0.7	68	3064	979812	2898	0.054	0.347	0.01733
0.7	34	4824	471164	4602	0.046	0.251	0.01252
0.7	17	7743	220080	7195	0.071	0.176	0.00880
0.7	8	12714	90100	11613	0.087	0.117	0.00585
0.7	4	19706	36458	17544	0.110	0.081	0.00403
0.7	2	31100	12658	25027	0.195	0.058	0.00289

Table 49 - Coverage Analysis Results, Estimator $\hat{\sigma}^2_{BM}$ on M/H_2/1 (Cx2=50) Queue

ρ	Batch Size	N_r	$\bar{l}_r$	N_a	p_d	$\bar{c}$	Δ_c
0.8	10400	2257	36281300	2017	0.106	0.959	0.01600
0.8	5200	1676	30088000	1518	0.094	0.918	0.01383
0.8	2600	1120	21876800	1011	0.097	0.851	0.02198
0.8	1300	706	13656800	663	0.061	0.745	0.03318
0.8	650	1066	7520330	1019	0.044	0.602	0.03007
0.8	325	1910	3952350	1766	0.075	0.465	0.02327
0.8	162	3029	1936500	2890	0.046	0.347	0.01736
0.8	81	4973	920328	4563	0.082	0.252	0.01260
0.8	40	7821	418388	7115	0.090	0.178	0.00889
0.8	20	11933	181482	10914	0.085	0.124	0.00617
0.8	10	19178	72624	16829	0.122	0.084	0.00419
0.8	5	31214	25827	26961	0.136	0.054	0.00270
0.8	2	47899	6472	42395	0.115	0.035	0.00175

Table 50 - Coverage Analysis Results, Estimator $\hat{\sigma}^2_{BM}$ on M/H_2/1 (Cx2=50) Queue

ρ	Batch Size	N_r	$\bar{l}_r$	N_a	p_d	$\bar{c}$	Δ_c
0.9	44000	2118	133395428	1887	0.109065	0.954	0.01478
0.9	22000	1573	110754285	1421	0.096631	0.919	0.01735

Table 51 - Coverage Analysis Results, Estimator $\hat{\sigma}^2_{BM}$ on M/H_2/1 (Cx2=50) Queue